

# Accessing and transforming data

SEA-EU course on Marine Data Literacy  
27 October 2021



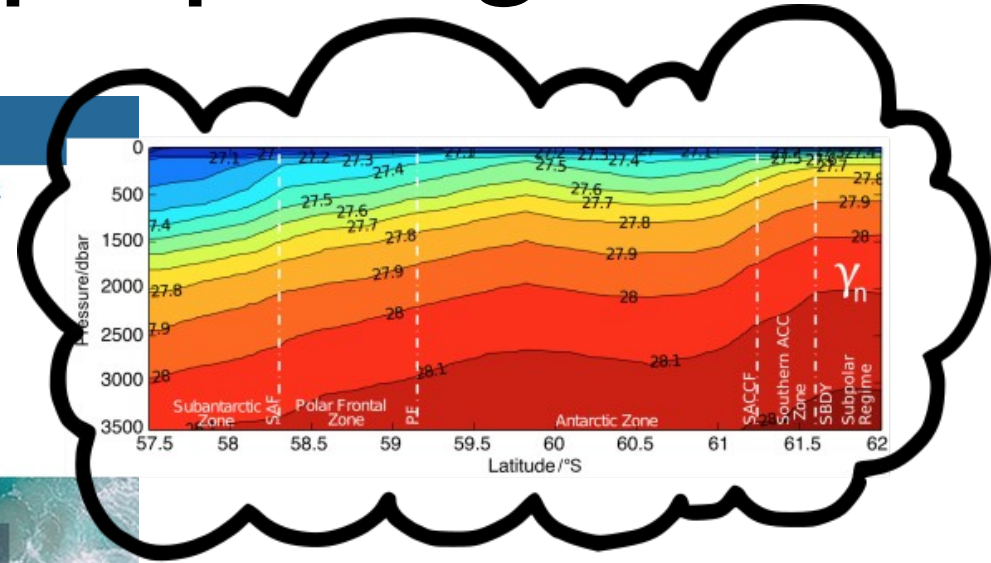
Sally Close, Université de Bretagne Occidentale

# Motivation: the practical aspects of acquiring and preparing data

Implemented by [Mercator Ocean International](#) as part of the [Copernicus Programme](#)



The Copernicus Marine Service logo features the European Union flag on the left, followed by the word "Copernicus" in a blue sans-serif font with "Europe's eyes on Earth" in smaller text below it. Below this is the "Copernicus Marine Service" logo, which includes a stylized blue fish icon and the text "Copernicus Marine Service" in blue.



## Copernicus Marine Service

Providing free and open marine data and services to enable marine policy implementation, support Blue growth and scientific innovation.

[Access Data >](#)

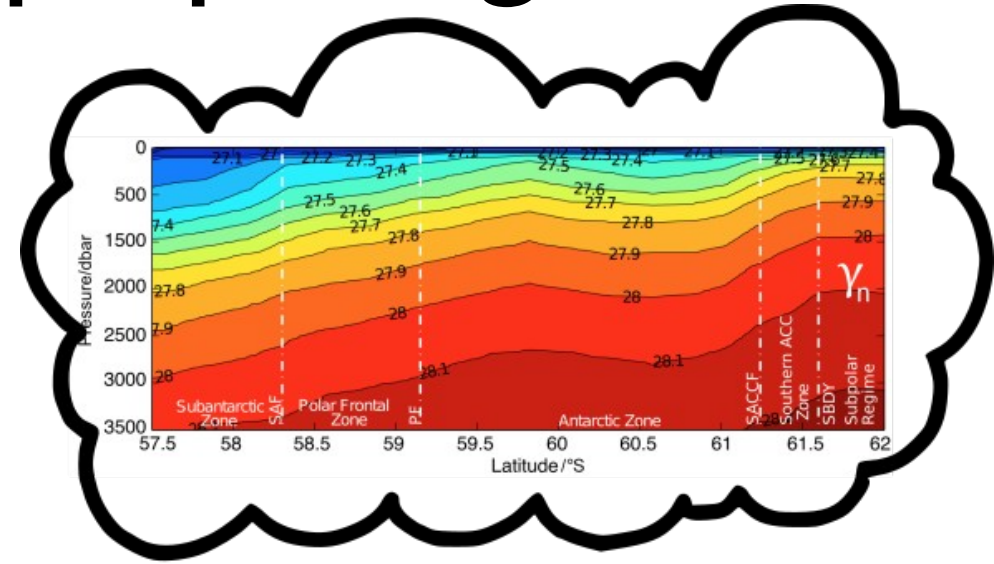
DATA	EXPERTISE
<b>OCEAN PRODUCTS</b> A robust ocean data catalogue, to	<b>OCEAN STATE REPORT</b> Extensive annual analysis on the



Navigation icons for the Copernicus Marine Service website, including a search icon, a home icon, and a user profile icon.

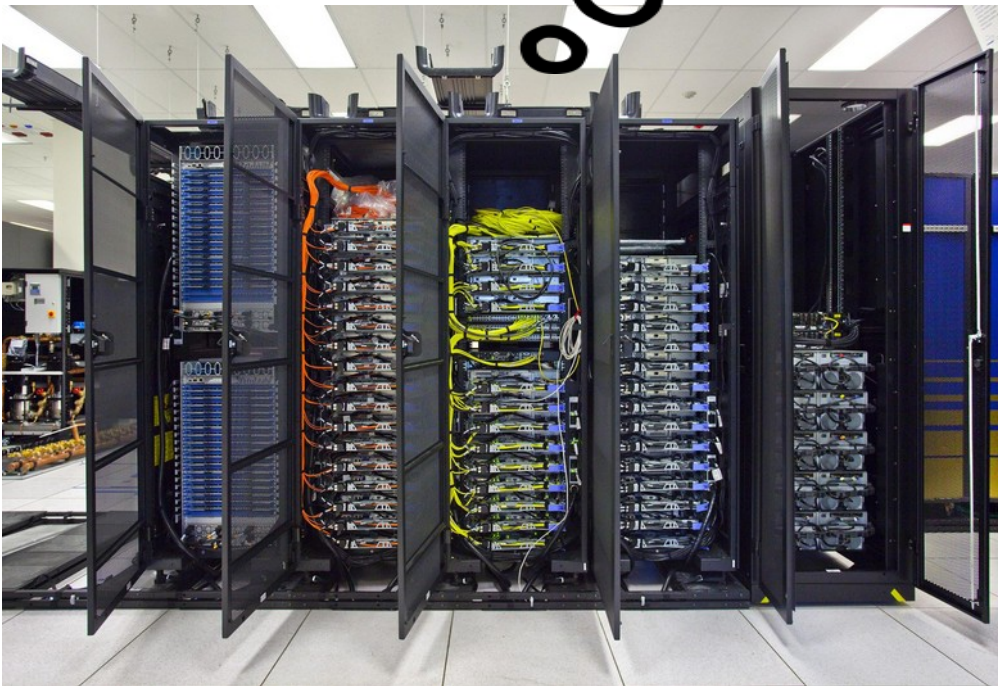
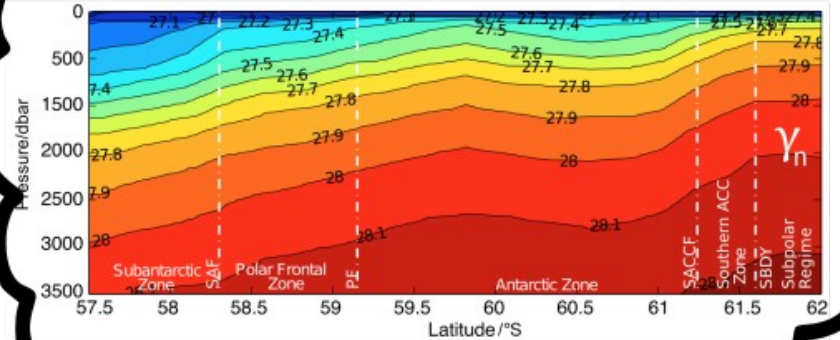


# Motivation: the practical aspects of acquiring and preparing data



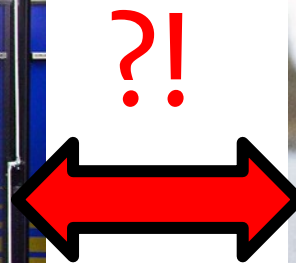
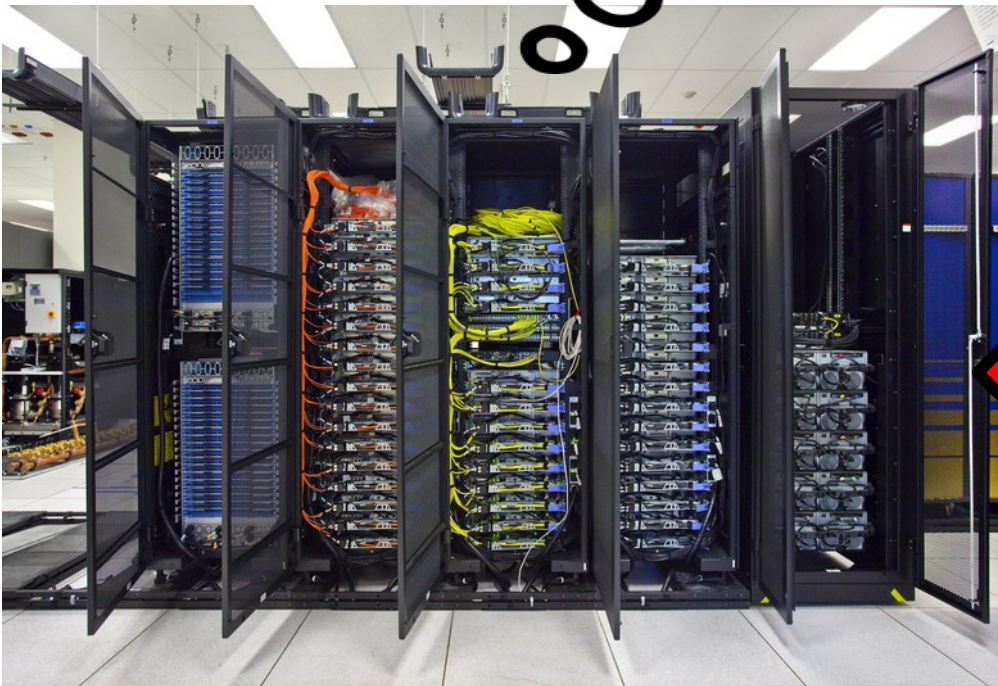
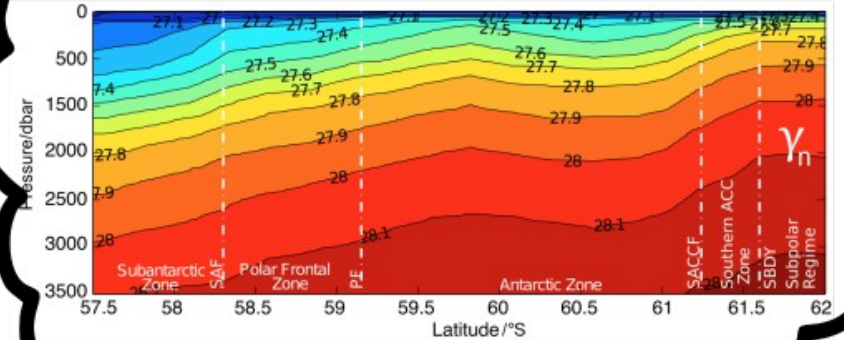
# Motivation: the practical aspects of acquiring and preparing data

1011100010101010  
1000001010111100  
01010101111100...



# Motivation: the practical aspects of acquiring and preparing data

1011100010101010  
1000001010111100  
01010101111100...



# A short, selected historical overview of oceanographic measurements:

- From a practical perspective, it is much more difficult to measure the ocean than the atmosphere.
- Oceanography is quite a young science, and for many years, the only way to take measurements was to go out on a ship with an instrument.
- This severely limited our knowledge of many aspects of the ocean and its role in the climate system. For example, the first direct estimates of ocean heat transport were only made in 1982!
- It is only very recently that we have started to have access to the large amounts of data that are available today...



Early weather balloon: photo from wikipedia

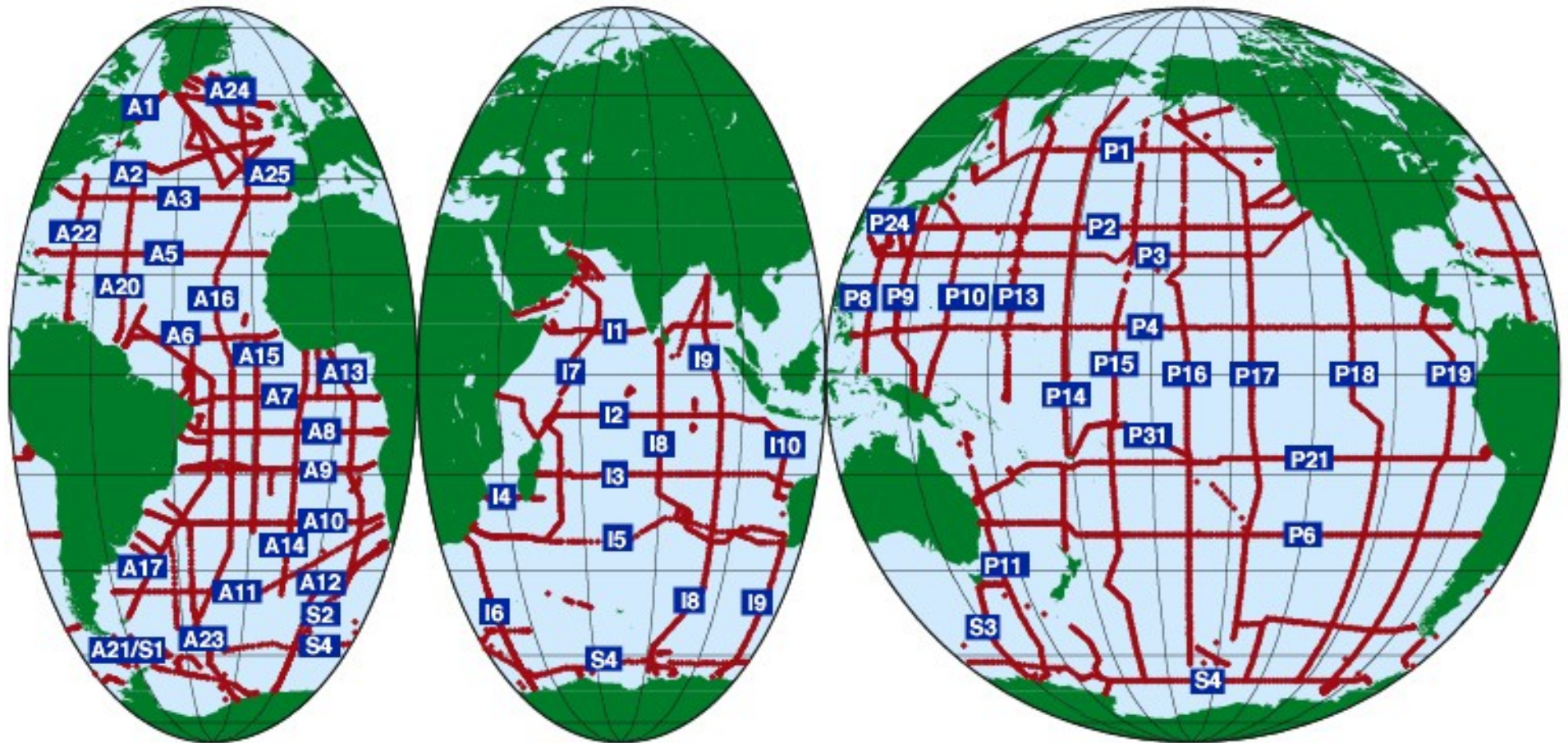


Photo from Scripps Institute of Oceanography: Roger Revelle taking a water sample in the 1930s

A modern CTD rosette: image from [lfremer](#)



# The WOCE “one time” measurement programme (1990–1998)





# Development of autonomous profiling floats:



Early 1950s: John Swallow invents the acoustically-tracked float (constructed here from scaffolding and tubing and Royal Navy sound sources that allowed the float to be located a few km away from a ship)

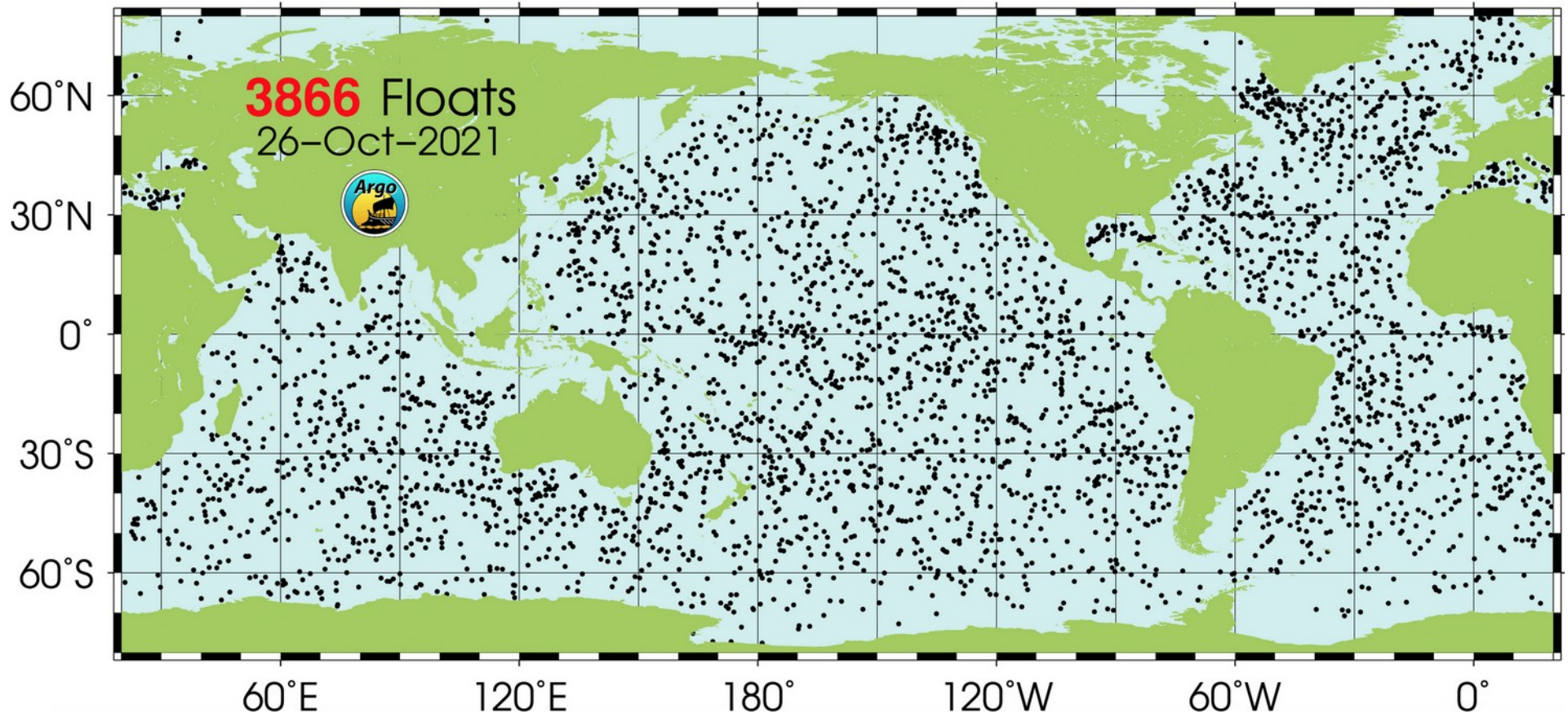
Image from: *Physical Oceanography, Developments Since 1950*; Ed. Jochum, Murtugudde



Image from the Ocean Carbon and Biogeochemistry program:  
<https://www.us-ocb.org/>

Late 1990s / early 2000s: development of the Argo programme – autonomous floats measuring temperature, salinity and depth are deployed worldwide by an alliance of countries. The target number of 3000 floats was achieved in October 2007.

# Present-day Argo float coverage



Modern floats are capable of measuring biogeochemical variables, and some are even equipped to survive in the polar regions:



Image from the Argo program: <https://argo.ucsd.edu/>

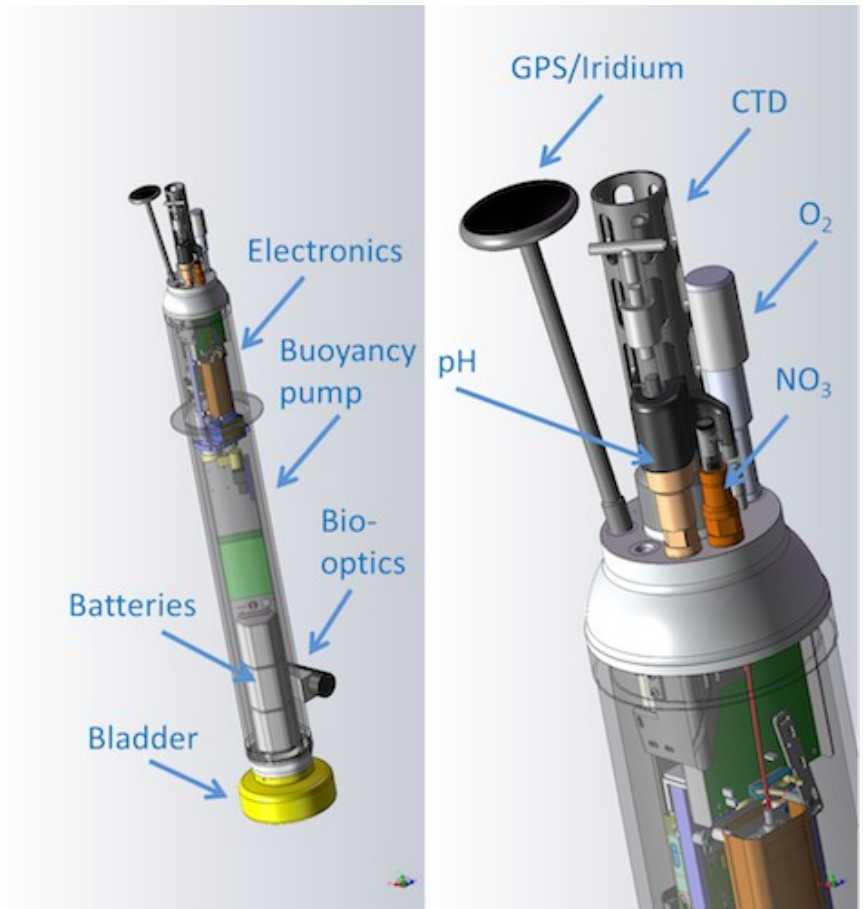


Image from the Ocean Carbon and Biogeochemistry program: <https://www.us-ocb.org/>

- The Argo programme has revolutionised our access to in situ ocean data.
- Satellite data has simultaneously given us unprecedented information about the ocean surface.
- Numerical ocean models have undergone similar radical improvements, notably in terms of spatial resolution
- With these developments, the amount of available data has increased dramatically over recent decades. This is reflected in how we access and work with data

# The main questions for today's lecture

- How does data get from somewhere else on the Internet to you?
- What do you need to take into account when retrieving the data?

# The main questions for today's lecture

- How does data get from somewhere else on the Internet to you?
  - This will depend on (amongst other things):
    - What type of data you want to retrieve
    - Where the data is being stored
    - Where you want to process it

# The main questions for today's lecture

- What do you need to take into account when retrieving the data?
- This will depend on (amongst other things):
  - How big the data set is that you are trying to retrieve
  - The format of the data (Gridded? A time series? In a text format? A binary?)
  - Whether you want to retrieve all of the data, or only part of it (Certain variables only? Certain regions only?)

# The plan for today:

	Topic:	What is it ?
1:	Web scraping	A generic method of collecting information from a web page
2:	Data wrangling	The process of transforming data into a format in which you can use it easily to perform your required calculations
3:	API	A software “messenger” designed to let two computers (or pieces of software) talk to one another more easily
4:	OpenDAP	A type of API that is widely used in the Earth sciences. It lets us tell data servers what we want when we interact with them
5:	Cloud computing	A way of performing calculations or storing data on machines that are owned by somebody else without transferring data directly to your computer
6:	The Zarr format	A format of data that is native to cloud computing and designed to deal with very large data sets
7:	Pangeo, Binder, Google Colab	Tools to allow you to work and compute in the cloud

# Web scraping

- Web scraping = extracting information from websites by processing the HTML code of the website
- How does it work?
  1. Write a program / use a piece of software to communicate with a website and request data
  2. The website returns the data (usually the HTML code of the webpage + any other associated files)
  3. The program transforms the data to extract the information that you wanted
  4. Optional: the program stores the information and moves on to a new webpage



# Web scraping

- Some reasons that you might use web scraping:
  - You want to extract a lot of information (easier / less likely to have errors than with manual extraction)
  - You want to combine data from multiple websites to make a new data set
  - You want to process / archive the data
  - There isn't a better way to access the data...
- Some reasons that you might **not** use web scraping:
  - If the owner of the website has specified that they don't want you to scrape their data (robots.txt)
  - There is a better way to access the data (e.g. an API – coming up next!)

# Example: high tide information



high tide in brest



[All](#)

[Images](#)

[Maps](#)

[Videos](#)

[News](#)

[More](#)

[Tools](#)

About 621,000 results (0.68 seconds)

Today's tide times for Brest: Friday 22 October 2021

Tide	Time (CEST)& Date	Height
Low Tide	00:50 AM(Fri 22 October)	3.97 ft (1.21 m)
High Tide	<b>6:44 AM</b> (Fri 22 October)	22.61 ft (6.89 m)
Low Tide	1:05 PM(Fri 22 October)	4.1 ft (1.25 m)
High Tide	6:59 PM(Fri 22 October)	22.51 ft (6.86 m)

<https://www.tide-forecast.com> › [Brest-France](#) › [tides](#) › latest

[Tide Times and Tide Chart for Brest - Tide Forecast](#)



About featured snippets



Feedback

# Example: high tide information

Google high tide in brest

Google high tide in gdansk

About 621,000

Today's tide

Tide

Low Tide

High Tide

Low Tide

High Tide

Today's tide times for Gdansk: Saturday 23 October 2021

<https://www.tide-forecast.com>

Tide Times:






Tide	Time (CEST)& Date	Height
High Tide	1:35 AM(Sat 23 October)	0.1 ft (0.03 m)
Low Tide	8:09 AM(Sat 23 October)	0.03 ft (0.01 m)
High Tide	<b>2:36 PM(Sat 23 October)</b>	0.1 ft (0.03 m)
Low Tide	8:12 PM(Sat 23 October)	0.03 ft (0.01 m)

<https://www.tide-forecast.com> > Gdansk > tides > latest







[Tide Times and Tide Chart for Gdansk - Tide Forecast](#)

About featured snippets • Feedback

# Example: high tide information

Google  high tide in valletta ×   ×   Tools

About 621,000

Today's tide |  All  Images  Maps  News  Videos  More Tools

**Tide**

About 269,000 results (0.63 seconds)

Low Tide

High Tide Find the times of the next tides in Valletta in Malta on this page. m)

Low Tide ... 1 m)


High Tide Monday 25 October 2021. m)

<https://www.tide-times.com/>



**Tide Time:**

Tide	hour	tidal range
high tide	07:08	1.3ft
low tide	12:12	1ft
high tide	19:15	1.3ft

1 more row

<https://www.seatemperatu.re> > Europe > Malta > Valletta 

[Tide in Valletta: Full 7-day tide schedule - SeaTemperatu.re](https://www.seatemperatu.re)

 About featured snippets •  Feedback

# Example: high tide information

Google search for "high tide in brest".

About 621,000 results (0.68 seconds)

Today's tide times for Brest: Friday 22 October 2021

Tide	Time (CEST)& Date	Height
Low Tide	00:50 AM(Fri 22 October)	3.97 ft (1.21 m)
High Tide	<b>6:44 AM(Fri 22 October)</b>	22.61 ft (6.89 m)
Low Tide	1:05 PM(Fri 22 October)	4.1 ft (1.25 m)
High Tide	6:59 PM(Fri 22 October)	22.51 ft (6.86 m)

Google search for "high tide in gdansk".

About 616,000 results (0.51 seconds)

Today's tide times for Gdansk: Saturday 23 October 2021

Tide	Time (CEST)& Date	Height
High Tide	1:35 AM(Sat 23 October)	0.1 ft (0.03 m)
Low Tide	8:09 AM(Sat 23 October)	0.03 ft (0.01 m)
High Tide	<b>2:36 PM(Sat 23 October)</b>	0.1 ft (0.03 m)
Low Tide	8:12 PM(Sat 23 October)	0.03 ft (0.01 m)

Google search for "high tide in valletta".

About 269,000 results (0.63 seconds)

Find the times of the next tides in Valletta in Malta on this page.  
...  
Monday 25 October 2021.

Tide	hour	tidal range
high tide	<b>07:08</b>	<b>1.3ft</b>
low tide	12:12	1ft
high tide	19:15	1.3ft

[1 more row](#)

<https://www.seatemperatu.re> > Europe > Malta > Valletta > [Tide in Valletta: Full 7-day tide schedule - SeaTemperatu.re](#)

s://www.tide-forecast.com > Gdansk > tides > latest > [le Times and Tide Chart for Gdansk - Tide Forecast](#)

About featured snippets • Feedback

- We could use web scraping to automate the process of finding the high tide times for multiple locations
- To do this, we would need to look at the HTML code of the webpage, identify the part that corresponds to the table, and extract this

# robots.txt

- The website owner needs to agree that it's ok for you to extract information from their webpage
- They might not agree. This could be because:
  - 1.They don't want other people to store their information
  - 2.Scraping can put a lot of demand on the server, particularly when done badly
- The most common way to specify whether it's ok or not for you to scrape a website is by using a file called robots.txt. This is a list of instructions that tells the software (e.g. browser, web crawler, ...) that is accessing the website what it is allowed to do
- If you violate the owner's wishes, you may find yourself blocked from the website!

# Web scraping summary

- Web scraping is a generic way to access information from a website by retrieving and processing the HTML code of the web page and associated information
- It is widely used in data science and is very flexible and relatively efficient, but...
- ...it also has a number of disadvantages:
  - It can demand quite a lot of effort to set up
  - If the website changes, you need to rewrite your program
  - You need to work out not only how to retrieve the data, but also how to process it
  - Not all websites allow it

# Data wrangling

- Data wrangling is the process of transforming your input data into a format in which you can analyse it
- This can take several potential forms:
  - Removing headers or extra text
  - Dealing with missing values
  - Reading in tabular data and combining columns or rows
  - Combining multiple data sources



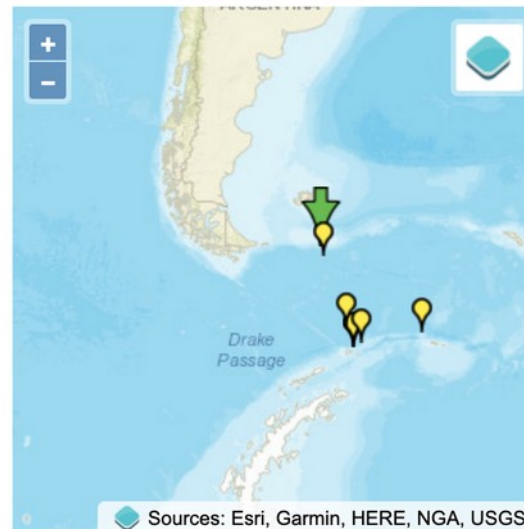
# Example: combining sea level measurements with bottom pressure recorder data

## Drake Passage North

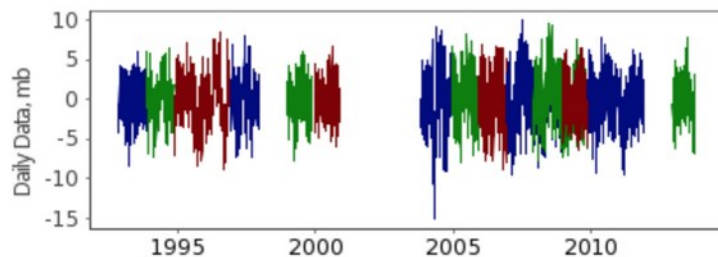
### Information

Latitude: -54.9464  
Longitude: -58.3405  
Start of first deployment: 1992-11-09  
End of last deployment: 2013-10-05  
Ocean region: 1.4

**Green arrow:** Current Bottom Pressure Location  
**Yellow Marker:** Other Bottom Pressure Location



### All Data



↓ All Best Hourly  
↓ All Best Daily

Data available at the Permanent Service for Mean Sea Level:

[https://www.psmsl.org/data/bottom\\_pressure/locations/72.php](https://www.psmsl.org/data/bottom_pressure/locations/72.php)

We can see that there are gaps in the data record

# The list of available files for this instrument:

## Deployments (best channel from each deployment)

[Go to page with all data from this deployment.](#)

Start	End	Latitude	Longitude	Depth	Data	Metadata
1992-11-09	1993-11-22	-54.9423	-58.3932	1010	DPN9293_DQ41086: <a href="#">Hourly</a> , <a href="#">Daily</a> , <a href="#">Tide</a>	<a href="#">Metadata</a>
1993-11-21	1994-11-15	-54.943	-58.3918	1007	DPN9394_DQ41083: <a href="#">Hourly</a> , <a href="#">Daily</a> , <a href="#">Tide</a>	<a href="#">Metadata</a>
1994-11-11	1996-11-14	-54.9442	-58.3852	1057	DPN9496_DQ44935: <a href="#">Hourly</a> , <a href="#">Daily</a> , <a href="#">Tide</a>	<a href="#">Metadata</a>
1996-11-23	1997-12-19	-54.944	-58.3837	1077	DPN9697_QD49187: <a href="#">Hourly</a> , <a href="#">Daily</a> , <a href="#">Tide</a>	<a href="#">Metadata</a>
1998-12-07	1999-11-11	-54.9427	-58.3568	1198	DPN9899_DQ46251: <a href="#">Hourly</a> , <a href="#">Daily</a> , <a href="#">Tide</a>	<a href="#">Metadata</a>
1999-12-08	2000-11-23	-54.9427	-58.3583	1203	DPN9900_DQ68484: <a href="#">Hourly</a> , <a href="#">Daily</a> , <a href="#">Tide</a>	<a href="#">Metadata</a>
2003-10-30	2004-12-04	-54.9432	-58.3568	1148	DPN0304_DQ68485: <a href="#">Hourly</a> , <a href="#">Daily</a> , <a href="#">Tide</a>	<a href="#">Metadata</a>
2004-12-18	2005-12-07	-54.9432	-58.3568	1090	DPN0405_DQ43513: <a href="#">Hourly</a> , <a href="#">Daily</a> , <a href="#">Tide</a>	<a href="#">Metadata</a>
2005-12-06	2006-12-08	-54.9432	-58.3568	1090	DPN0506_DQ68483: <a href="#">Hourly</a> , <a href="#">Daily</a> , <a href="#">Tide</a>	<a href="#">Metadata</a>
2006-12-08	2008-12-13	-54.9432	-58.3568	1218	DPN0608_DQ68485: <a href="#">Hourly</a> , <a href="#">Daily</a> , <a href="#">Tide</a>	<a href="#">Metadata</a>
2007-11-27	2009-11-19	-54.9437	-58.3767	1172	DPN0709_DQ68483: <a href="#">Hourly</a> , <a href="#">Daily</a> , <a href="#">Tide</a>	<a href="#">Metadata</a>
2008-12-11	2009-11-19	-54.9432	-58.3568	1227	DPN0809_DQ46267: <a href="#">Hourly</a> , <a href="#">Daily</a> , <a href="#">Tide</a>	<a href="#">Metadata</a>
2009-11-10	2011-11-29	-54.9432	-58.3568	1227	DPN0911_DQ68489: <a href="#">Hourly</a> , <a href="#">Daily</a> , <a href="#">Tide</a>	<a href="#">Metadata</a>
2012-12-14	2013-10-05	-54.9901	-57.96718	1086	DPN1215_DQ68489: <a href="#">Hourly</a> , <a href="#">Daily</a> , <a href="#">Tide</a>	<a href="#">Metadata</a>

# Inside the data file...

```
DPN9293_DQ41086_hrp.txt
Ocean Bottom Pressure Record distributed by PSMSL - Hourly Mean Data

Location name:           Drake Passage North
Deployment and Channel name: DPN9293_DQ41086
Latitude (degrees North): -54.9423
Longitude (degrees East): -58.3932
Data collected by:      National Oceanography Centre

Columns:
1: An integer count number
2: Flag for bad, missing data, or interpolated data. 0 means good data
3: Year
4: Day in year
5: Hour in day (UTC)
6: Pressure in millibars (hPa), approximately equivalent to 1 cm water
7: Tidal predictions in millibars
8: Estimated drift in millibars
9: Total residuals (column 6 - column 7 - column 8)

Recno Fl Year Day Hour Tot Pr Tide Drift Resid
..... 1 1 1992 314 14.500 -999.999 806.837 -999.999 -999.999
2 1 1992 314 15.500 -999.999 808.047 -999.999 -999.999
3 1 1992 314 16.500 -999.999 820.632 -999.999 -999.999
4 1 1992 314 17.500 -999.999 841.594 -999.999 -999.999
5 1 1992 314 18.500 -999.999 865.749 -999.999 -999.999
6 1 1992 314 19.500 -999.999 886.951 -999.999 -999.999
7 1 1992 314 20.500 -999.999 900.212 -999.999 -999.999
8 1 1992 314 21.500 -999.999 903.095 -999.999 -999.999
9 1 1992 314 22.500 -999.999 895.730 -999.999 -999.999
10 1 1992 314 23.500 -999.999 880.534 -999.999 -999.999
11 1 1992 315 0.500 -999.999 862.112 -999.999 -999.999
12 1 1992 315 1.500 -999.999 846.298 -999.999 -999.999
13 1 1992 315 2.500 -999.999 838.081 -999.999 -999.999
14 1 1992 315 3.500 -999.999 839.984 -999.999 -999.999
15 1 1992 315 4.500 -999.999 851.872 -999.999 -999.999
16 1 1992 315 5.500 -999.999 871.233 -999.999 -999.999
17 1 1992 315 6.500 -999.999 893.238 -999.999 -999.999
18 1 1992 315 7.500 -999.999 911.600 -999.999 -999.999
19 1 1992 315 8.500 -999.999 920.757 -999.999 -999.999
20 1 1992 315 9.500 -999.999 917.827 -999.999 -999.999
```

Before we can make a plot with this, or use the measurements for an analysis we need to...

Remove all this text (metadata) at the beginning

Separate out the variables in each of the columns

Decide what to do about the missing values (-999.999)

And we need to do this for every file...

The website also links to another file containing metadata associated with the data file:

## **Deployment DPN9293\_2013**

(Drake Passage North)

### **Location**

Latitude: -54.9423

Longitude: -58.3932

Depth: 1010 m

Ocean region: 1.4 - South Atlantic Ocean

### **Time Span**

Start Date: 1992-11-09

End Date: 1993-11-22

### **Notes**

An offset of 103 bar was removed from the raw pressure data.

### **Channels**

#### **DPN9293\_2013\_QT2**

Parameter: temperature

Sensor Model: Quartz

Metadata is descriptive extra information (about data quality, processing that has been performed already, co-ordinates that are associated with the data...) that the data provider thinks you might need to know

The metadata might contain information that is relevant to your analysis or it might not. You might not even decide to keep it.

# Wrangling the data file...

```
DPN9293_DQ41086_hrp.txt
Ocean Bottom Pressure Record distributed by PSMSL - Hourly Mean Data

Location name:           Drake Passage North
Deployment and Channel name: DPN9293_DQ41086
Latitude (degrees North): -54.9423
Longitude (degrees East): -58.3932
Data collected by:      National Oceanography Centre

Columns:
1: An integer count number
2: Flag for bad, missing data, or interpolated data. 0 means good data
3: Year
4: Day in year
5: Hour in day (UTC)
6: Pressure in millibars (hPa), approximately equivalent to 1 cm water
7: Tidal predictions in millibars
8: Estimated drift in millibars
9: Total residuals (column 6 - column 7 - column 8)

Recno Fl Year Day Hour Tot Pr Tide Drift Resid
.....
1 1 1992 314 14.500 -999.999 806.837 -999.999 -999.999
2 1 1992 314 15.500 -999.999 808.047 -999.999 -999.999
3 1 1992 314 16.500 -999.999 820.632 -999.999 -999.999
4 1 1992 314 17.500 -999.999 841.594 -999.999 -999.999
5 1 1992 314 18.500 -999.999 865.749 -999.999 -999.999
6 1 1992 314 19.500 -999.999 886.951 -999.999 -999.999
7 1 1992 314 20.500 -999.999 900.212 -999.999 -999.999
8 1 1992 314 21.500 -999.999 903.095 -999.999 -999.999
9 1 1992 314 22.500 -999.999 895.730 -999.999 -999.999
10 1 1992 314 23.500 -999.999 880.534 -999.999 -999.999
11 1 1992 315 0.500 -999.999 862.112 -999.999 -999.999
12 1 1992 315 1.500 -999.999 846.298 -999.999 -999.999
13 1 1992 315 2.500 -999.999 838.081 -999.999 -999.999
14 1 1992 315 3.500 -999.999 839.984 -999.999 -999.999
15 1 1992 315 4.500 -999.999 851.872 -999.999 -999.999
16 1 1992 315 5.500 -999.999 871.233 -999.999 -999.999
17 1 1992 315 6.500 -999.999 893.238 -999.999 -999.999
18 1 1992 315 7.500 -999.999 911.600 -999.999 -999.999
19 1 1992 315 8.500 -999.999 920.757 -999.999 -999.999
20 1 1992 315 9.500 -999.999 917.827 -999.999 -999.999
```

The data can be transformed “by hand” the user:

- deletes the text at the beginning of the file
- then loads the data into a spreadsheet program to separate the columns
- Then decides whether to keep the missing values, or deletes them

Alternatively: all of the above can be done semi-automatically by using software (e.g. pandas library in python)

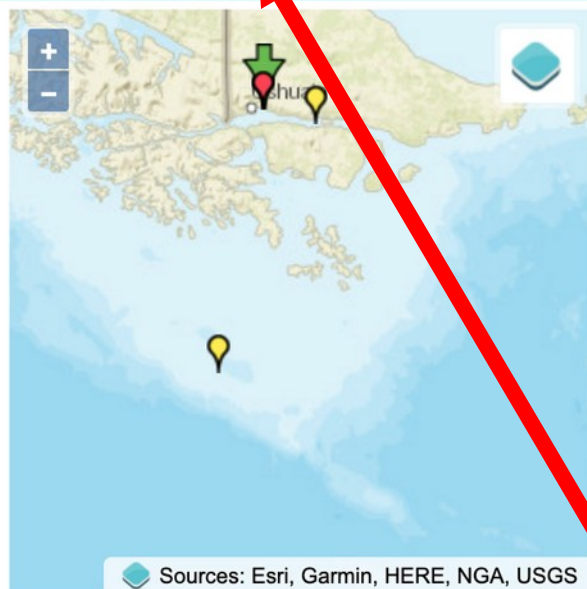
# Now let's look at the sea level file...

## USHUAIA II

WARNING: QCFLAG EXISTS. PLEASE READ THE DOCUMENTATION.

### Station Information

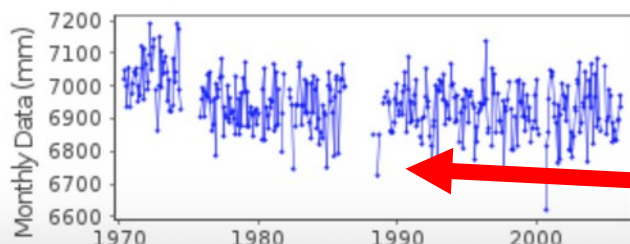
Station ID: 1271  
Latitude: -54.816667  
Longitude: -68.216667  
GLOSS ID: 181  
Coastline code: 860  
Station code: 2  
Country: ARGENTINA  
Time span of data: 1970 – 2006  
Completeness (%): 81  
Date of last update: 12 Nov 2007



**Green Arrow:** Current Station  
**Yellow Marker:** Neighbouring RLR Station  
**Red Marker:** Neighbouring Metric Station

Please note: In many cases, the station position in our database is accurate to only one minute. Thus, the tide gauge may not appear to be on the coast.

### Tide Gauge Data



[Link to larger image of monthly data plot.](#)  
[Download monthly mean sea level data.](#)

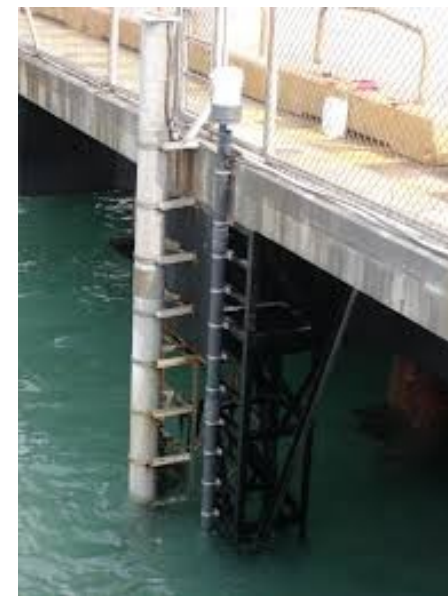


Image credit:  
[https://www.devb.gov.hk/filemanager/en/content\\_1044/20180107\\_08.html](https://www.devb.gov.hk/filemanager/en/content_1044/20180107_08.html)

Already we have several new features compared to the bottom pressure data page:

QC flag warning (best check meta data!)

Monthly, not hourly / daily data

And there are also some similarities: missing data

# Looking at the meta data...

## Station Documentation

[Link to RLR information.](#)

*Documentation added 1991-06-11*

Ushuaia II 860/002 RLR(1977) is 10.0m below Pilar MOP No 4377

*Documentation added 1993-07-12*

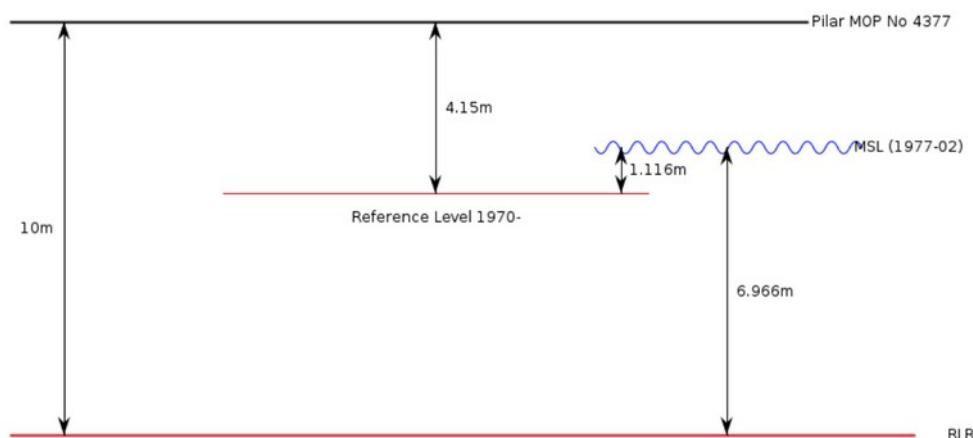
Evidence for possible datum shift following gap in 1970s.

*Documentation added 1999-08-20*

station is equipped with both float gauge and NGWLMS system. Last data 2006

## Revised Local Reference (RLR) Diagram for USHUAIA II

Station ID: 1271



If the image above appears blurry, or you would like to see a larger image, please view the [full-sized image of the RLR diagram](#).

## Datum information

Add 5.850m to all data values to refer to RLR

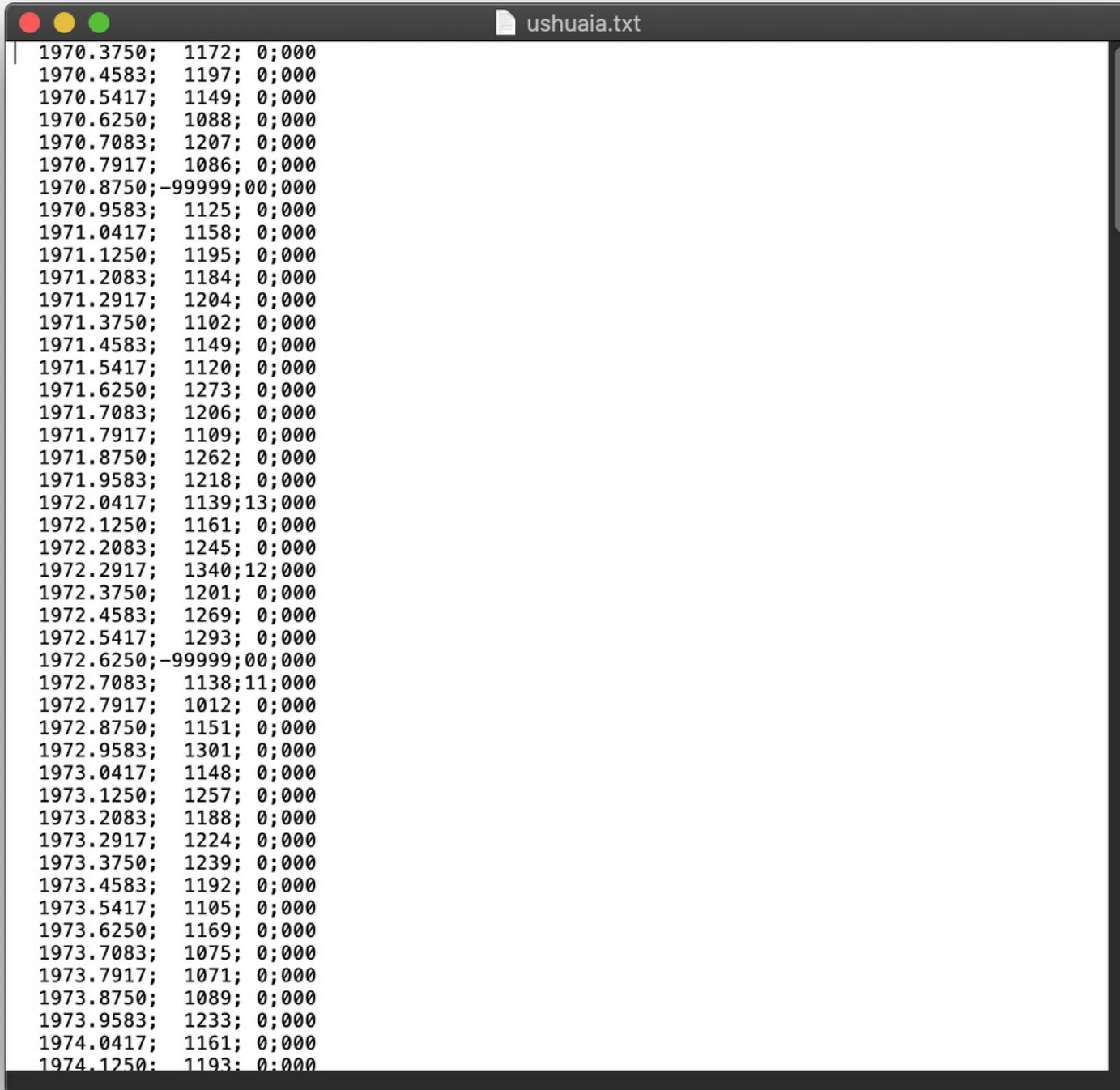
RLR is 10.000m below primary benchmark Pilar MOP No 4377

Here, the metadata tells us about some recommended pre-processing that we should perform if we want the data to be consistent with other instruments.

This could be important if we are combining multiple data sources.

→ even if you don't use the metadata, it's still a good idea to check it!

# Inside the data file...



```
1970.3750; 1172; 0;000
1970.4583; 1197; 0;000
1970.5417; 1149; 0;000
1970.6250; 1088; 0;000
1970.7083; 1207; 0;000
1970.7917; 1086; 0;000
1970.8750;-99999;00;000
1970.9583; 1125; 0;000
1971.0417; 1158; 0;000
1971.1250; 1195; 0;000
1971.2083; 1184; 0;000
1971.2917; 1204; 0;000
1971.3750; 1102; 0;000
1971.4583; 1149; 0;000
1971.5417; 1120; 0;000
1971.6250; 1273; 0;000
1971.7083; 1206; 0;000
1971.7917; 1109; 0;000
1971.8750; 1262; 0;000
1971.9583; 1218; 0;000
1972.0417; 1139;13;000
1972.1250; 1161; 0;000
1972.2083; 1245; 0;000
1972.2917; 1340;12;000
1972.3750; 1201; 0;000
1972.4583; 1269; 0;000
1972.5417; 1293; 0;000
1972.6250;-99999;00;000
1972.7083; 1138;11;000
1972.7917; 1012; 0;000
1972.8750; 1151; 0;000
1972.9583; 1301; 0;000
1973.0417; 1148; 0;000
1973.1250; 1257; 0;000
1973.2083; 1188; 0;000
1973.2917; 1224; 0;000
1973.3750; 1239; 0;000
1973.4583; 1192; 0;000
1973.5417; 1105; 0;000
1973.6250; 1169; 0;000
1973.7083; 1075; 0;000
1973.7917; 1071; 0;000
1973.8750; 1089; 0;000
1973.9583; 1233; 0;000
1974.0417; 1161; 0;000
1974.1250; 1193; 0;000
```

This file will be easier to process than the bottom pressure recorder file because it doesn't contain extra text

However:

- we need to look up what all the columns mean because there is no information in the file
- If we want to combine this with the other data, we need to make time averages for each month in the other data, because the time interval is not the same here



# Data wrangling

- Preparing data for use is often one of the most boring parts of analysis. But it needs to be done when dealing with raw data such as these examples
- If you have collected data by web scraping, you will almost certainly need to do some of this type of preparation before analysis
- Luckily, for many types of data, we don't have to do much preparation at all: data that is gridded (satellite observations, reanalysis products, model outputs, ...) are usually much easier to deal with!
- These gridded data are normally supplied in a format called NetCDF that allows us to store metadata with the data all in one single file. Our standard software tools in oceanography are well-equipped to deal with this data format, and you may not need to do any data wrangling at all if you're lucky!
- Almost all marine data centres will allow you to download data in NetCDF format. So, how do we get data from a data centre...?

# API: a way for computers to communicate with one another

- API stands for *Application Programming Interface*
- When you use software, or even an operating system, you interact with the user interface: this is how you tell the computer what you want it to do
- An API plays a similar role, but instead of allowing a human to interact with a machine, it allows two computers, or two pieces of software to interact with one another

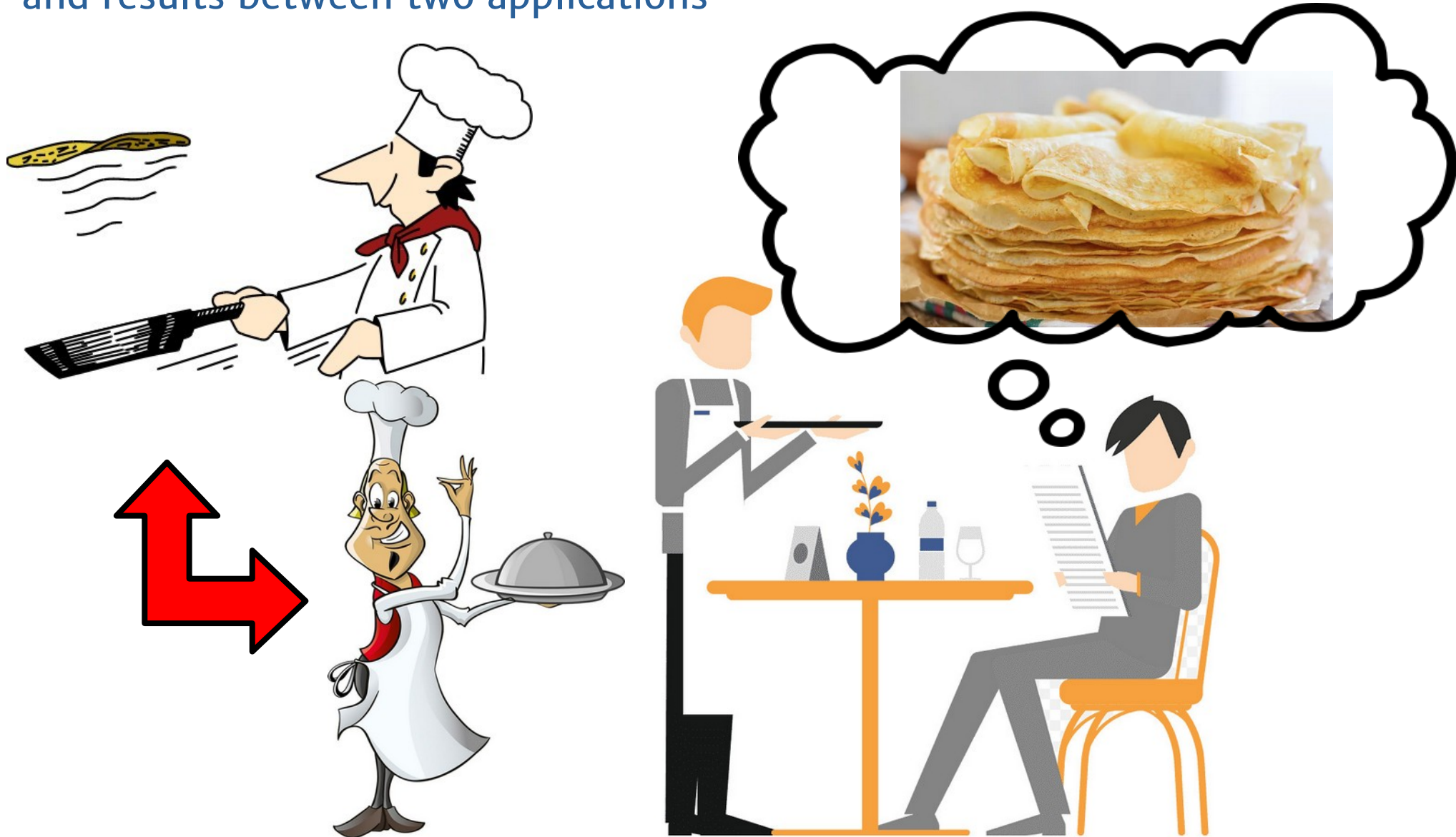
# API: the waiter analogy

When you order food in a restaurant, you do not prepare the food yourself directly or interact with the chef: **the waiter acts as the intermediary**, transmitting your wishes to the kitchen and returning with the results



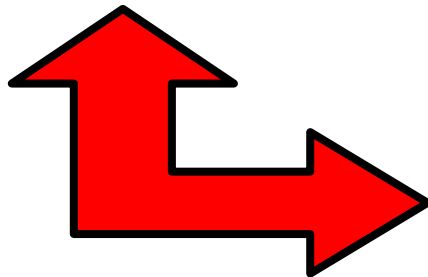
# API: the waiter analogy

The work of the API is similar to the work of the waiter: it carries requests and results between two applications



# API: the waiter analogy

The work of the API is similar to the work of the waiter: it carries requests and results between two applications

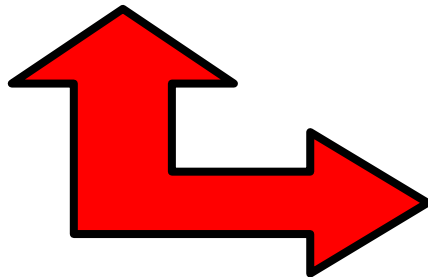
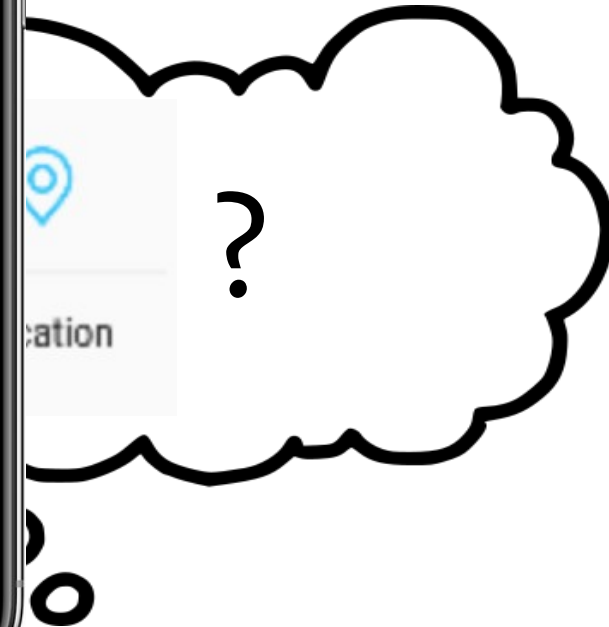
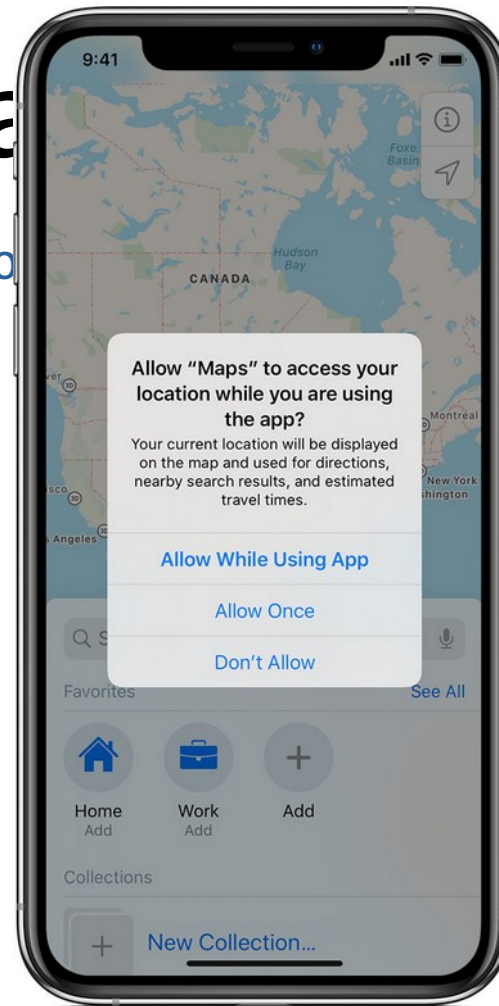


Google Maps

# API: the way technology

The work of the API is similar to the work of a waiter who carries requests and

carries requests and



Google Maps

# API for obtaining data

- There are many, many, many different types of API for many different purposes
- We are learning about them here because a certain type of API are the method that data providers use to allow users to select and download climate data: these are **web API**
- The user tells the web API what subset of the data they want, and the web API retrieves it from the server. **The user interacts with the web API indirectly: either through a webpage or programmatically (often through Python)**

# Why do we need API?

- Ocean / climate data are generally a lot more complicated than the standard information that we find on webpages
- They are multidimensional (up to 4D), often structured (gridded), may come with metadata, potentially large spatial scale and high temporal resolution
- Which region do you want to look at? (The whole globe? Europe? The Indian Ocean?) With what time resolution? (Hourly? Daily? Monthly? Yearly?) At what height / depth / pressure level? ...
- In many cases, it is unlikely that you need the full data set, and it would in any case not be manageable to work with



# Data centre API

- The various data centres that supply ocean and climate data have their own, individual API
- For example: ECMWF supply a python API library...  
<https://confluence.ecmwf.int/display/WEBAPI/Access+ECMWF+Public+Datasets>
- ... or also allow access via a web interface:  
<https://apps.ecmwf.int/datasets/data/interim-full-daily/levtype=sfc/>
- CMEMS use the same approach with a web catalogue:  
<https://resources.marine.copernicus.eu/products>
- Or a variety of different API access options:  
[https://help.marine.copernicus.eu/en/articles/4794731-which-apis-are-provided#h\\_c58ec72061](https://help.marine.copernicus.eu/en/articles/4794731-which-apis-are-provided#h_c58ec72061)

# Interacting with the API

Example from ECMWF:

## MARS request

```
retrieve,  
  stream=oper,  
  levtype=sfc,  
  param=165.128/41.128,  
  dataset=interim,  
  step=0,  
  grid=0.75/0.75,  
  time=00,  
  date=2013-09-01/to/2013-09-30,  
  type=an,  
  class=ei
```

## Python equivalence

```
#!/usr/bin/env python  
from ecmwfapi import ECMWFDataServer  
  
server = ECMWFDataServer()  
  
server.retrieve({  
    'dataset' : "interim",  
    'time'    : "00",  
    'date'    : "2013-09-01/to/2013-09-30",  
    'step'    : "0",  
    'type'    : "an",  
    'levtype' : "sfc",  
    'param'   : "165.128/41.128",  
    'grid'    : "0.75/0.75",  
    'target'  : "interim201309.grib"  
})
```

Hint: double-click

# Example of generating an API client script automatically on the CMEMS website

- [https://resources.marine.copernicus.eu/product-detail/GLOBAL\\_REANALYSIS\\_PHY\\_001\\_030/INFORMATION](https://resources.marine.copernicus.eu/product-detail/GLOBAL_REANALYSIS_PHY_001_030/INFORMATION)

Implemented by [Mercator Ocean International](#) as part of the [Copernicus Programme](#)



BETA



Home

Access Data

User Corner

Contact Us

INFORMATION

DOCUMENTATION

SERVICES

NOTIFICATIONS

Product identifier

GLOBAL\_REANALYSIS\_PHY\_001\_030

Overview

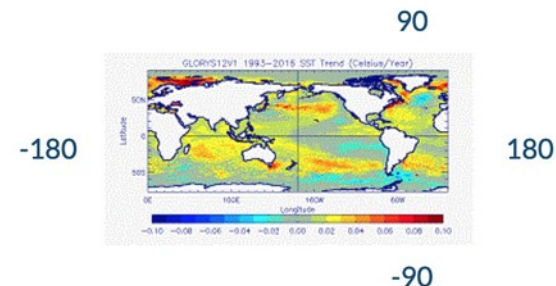
**Short description:**

The GLORYS12V1 product is the CMEMS global ocean eddy-resolving (1/12° horizontal resolution, 50 vertical levels) reanalysis covering the altimetry (1993 onward).

It is based largely on the current real-time global forecasting CMEMS system. The model component is the NEMO platform driven at surface by ECMWF ERA-Interim then ERA5 reanalyses for recent years. Observations are assimilated by means of a reduced-order Kalman filter. Along track altimeter data (Sea Level Anomaly),

Areas : global-ocean

Geographical coverage



Observation / Models

numerical-model



# API summary

- API are intermediaries between an application that the user interacts with and a target application
- Web API are widely used by data centres to allow users to access data either through web interfaces or programmatically, through scripts (normally via Python)
- These API permit us to select subsets of the data. This is important in marine science because the entire data set may be too large to work with, and provide much more information than we really need to solve our problem
- Different data centres have different API, but the way that we interact with them is often similar

# OPeNDAP



## OPeNDAP™













Advanced Software for Remote Data Retrieval

- OPeNDAP = Open-source Project for a Network Data Access Protocol
- OPeNDAP is an API that has been widely used in the Earth science community for many years: many marine data servers offer this particular method
- It is a generic tool, and is not used only within marine science
- It is well suited to ocean / climate data because it works with structured data and allows selective data retrieval, and interfaces with many commonly-used analysis tools (IDL, Python, R, CDO, MATLAB, Ferret, GrADS...)
- The term OPeNDAP refers to both the software framework and also to the community of users and developers

# Example #1: retrieving Argo profiles from the Coriolis OPeNDAP server

- <http://tdso.ifremer.fr/thredds/catalog/CORIOLIS-ARGO-GDAC-OBS/catalog.html>

**ifremer** Catalog <http://tdso.ifremer.fr/thredds/catalog/CORIOLIS-ARGO-GDAC-OBS/catalog.html>

Dataset	Size	Last Modified
 <a href="#">CORIOLIS-ARGO-GDAC-OBS</a>		--
 <a href="#">nmdis/</a>		--
 <a href="#">meds/</a>		--
 <a href="#">kordi/</a>		--
 <a href="#">kma/</a>		--
 <a href="#">jma/</a>		--
 <a href="#">incois/</a>		--
 <a href="#">csiro/</a>		--
 <a href="#">csio/</a>		--
 <a href="#">coriolis/</a>		--
 <a href="#">bodc/</a>		--
 <a href="#">aoml/</a>		--

# Example #2: retrieving a subset of Aquarius data from JPL

- <http://podaac-opendap.jpl.nasa.gov/opendap/allData/aquarius/>



## Contents of /allData/aquarius/L4/IPRC/v5/7day/2015/036/

Name	Last Modified	Size	DAP Response Links							Dataset Viewers	
<a href="#">SSS_OI_7D_20150362015042_V50.nc</a>	2018-07-11T20:29:42GMT	1042184	<a href="#">dax</a>	<a href="#">dds</a>	<a href="#">das</a>	<a href="#">info</a>	<a href="#">html</a>	<a href="#">rdf</a>	<a href="#">covjson</a>	<a href="#">file</a>	<a href="#">viewers</a>
<a href="#">SSS_OI_7D_20150362015042_V50.nc.md5</a>	2018-07-11T20:29:42GMT	65	-	-	-	-	-	-	-	-	-

THREDDS Catalog [XML](#)

Hyrax development sponsored by [NSF](#), [NASA](#), and [NOAA](#)

[OPeNDAP Hyrax \(1.16.0\)](#)  
[Documentation](#)

[Questions?](#) [Contact Support](#)

**ddx = XML version of  
DAS and DDS**

**dds = data set  
descriptor structure**

**das = data set  
attribute structure**

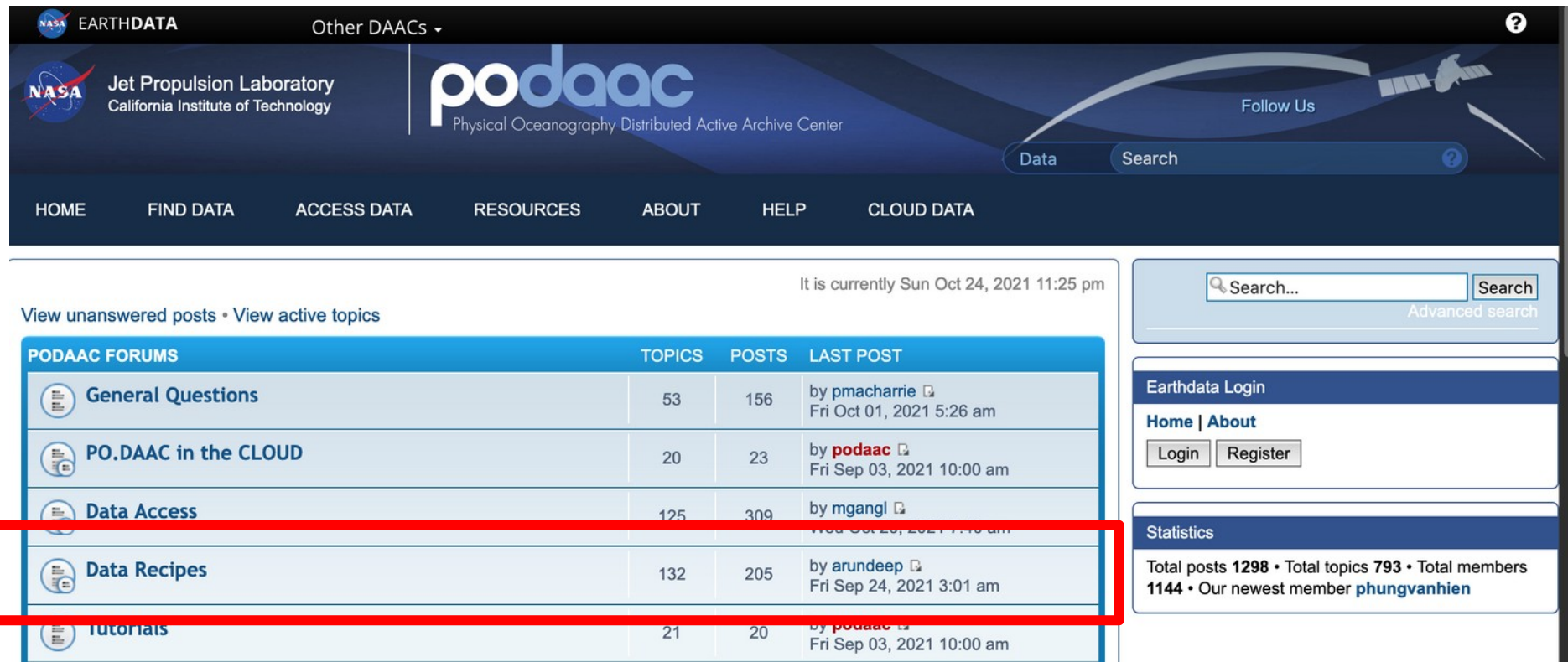
# THREDDS catalogues

- In the examples using the web interfaces that we have just seen, **the catalogue that we used to access the data is called a THREDDS catalogue**
- THREDDS = Thematic Real-time Environmental Distributed Data Services
- THREDDS data servers and catalogues are just interfaces to allow us to more easily explore and access scientific data. They **provide access to both metadata (useful descriptive information that lets us better use the data) and the data itself**
- For marine data access, THREDDS catalogues are widely used with the OPeNDAP API. THREDDS is the catalogue interface that you interact with, and OPeNDAP is the “waiter” that talks to the remote server








# Interacting with servers via OPeNDAP in your favourite software

- Instead of downloading the data files via a browser and then loading them into your favourite software, you can also load them directly into the software
- <https://www.opendap.org/allsoftware/other-sources>
- Many data centres provide examples to help you get started if you want to try this:



The screenshot shows the Podaac website interface. At the top, there is a header with the NASA Earthdata logo, the Jet Propulsion Laboratory logo, and the Podaac logo (Physical Oceanography Distributed Active Archive Center). Below the header is a navigation menu with links for HOME, FIND DATA, ACCESS DATA, RESOURCES, ABOUT, HELP, and CLOUD DATA. A search bar is located in the top right corner. The main content area features a forum table with the following data:

	TOPICS	POSTS	LAST POST
<a href="#">General Questions</a>	53	156	by pmacharrie  Fri Oct 01, 2021 5:26 am
<a href="#">PO.DAAC in the CLOUD</a>	20	23	by podaac  Fri Sep 03, 2021 10:00 am
<a href="#">Data Access</a>	125	309	by mgangl  Wed Oct 20, 2021 11:16 am
<a href="#">Data Recipes</a>	132	205	by arundeeep  Fri Sep 24, 2021 3:01 am
<a href="#">Tutorials</a>	21	20	by podaac  Fri Sep 03, 2021 10:00 am

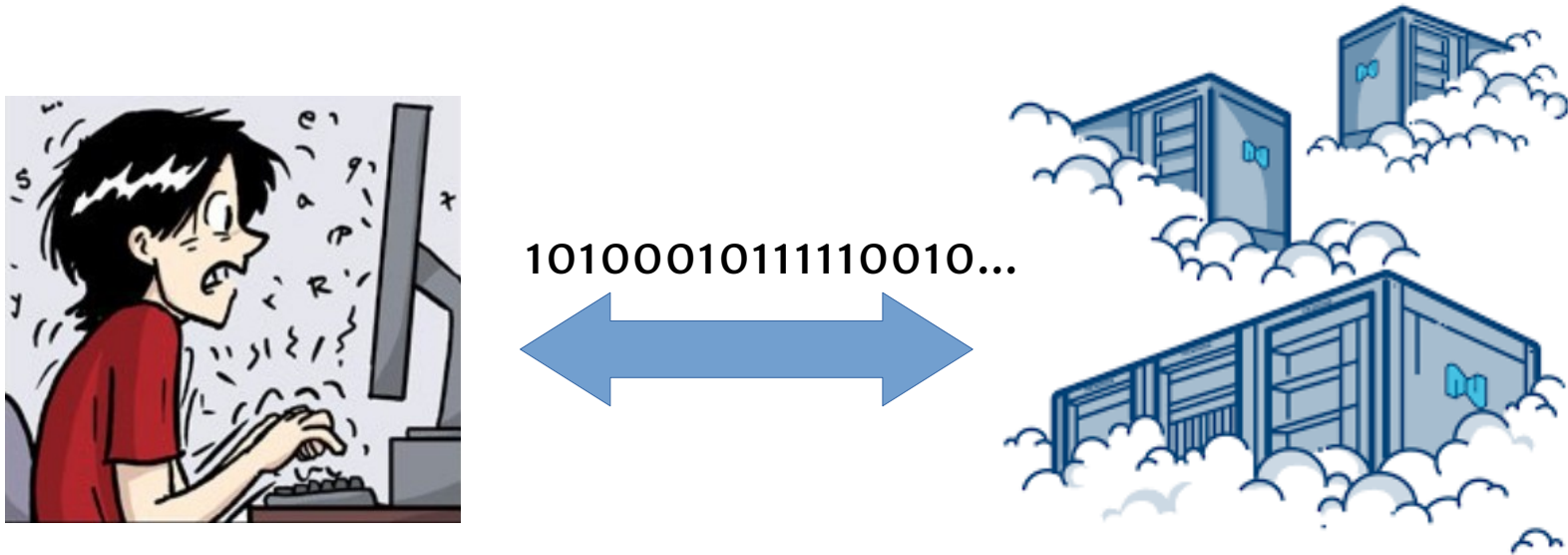
The 'Data Recipes' row is highlighted with a red rectangular box. To the right of the forum table, there is a search bar with the text 'Search...' and a 'Search' button. Below the search bar, there is a section for 'Earthdata Login' with links for 'Home' and 'About', and buttons for 'Login' and 'Register'. At the bottom right, there is a 'Statistics' section showing: 'Total posts 1298 • Total topics 793 • Total members 1144 • Our newest member phungvanhien'.

# OPeNDAP / THREDDS summary

- The OPeNDAP API is a protocol for accessing data that has been widely used in the Earth sciences for many years
- It is well adapted for the type of structured data that we use, because it allows us to select only subsets of the data, and also to examine and download the associated metadata that describes the data set
- THREDDS catalogues provide an easy way to navigate the available data via a website
- It is not necessary to download the data using a web browser: this can be done directly in most common data analysis software / programming languages

# Cloud computing

- So... what is cloud computing?



The essential idea is that you run a program/access data in such a way that to you, the user, it appears to be happening on your local machine/device, but in reality it is actually running/stored on an array of computers somewhere else

# Why has cloud computing become so popular?

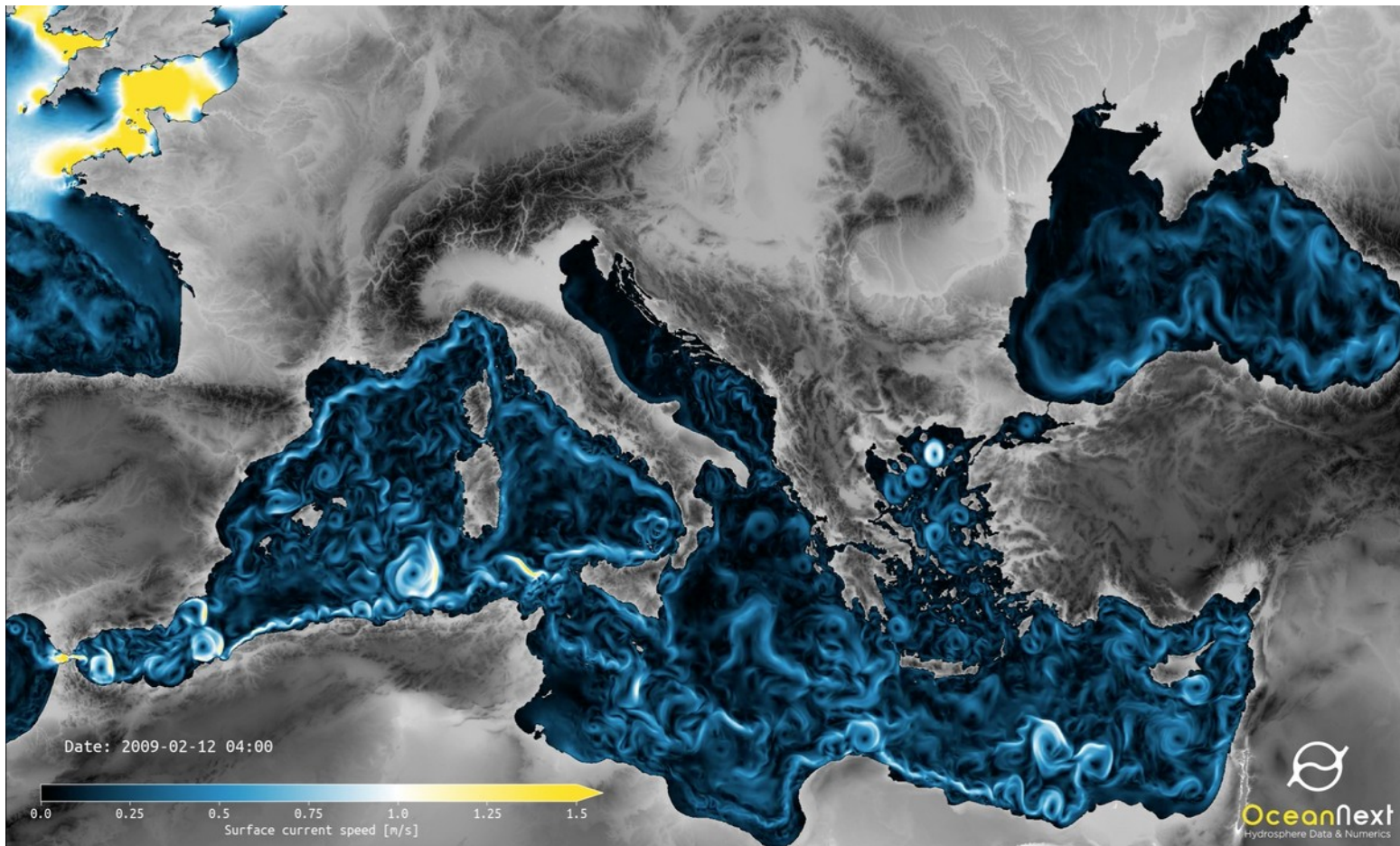
Lets start by considering the general case of e-commerce. There has been:

- A transformation in demand: many businesses have the same basic requirements (scalability, data storage and processing)
- A transformation in data transmission: data transfer speeds have increased dramatically with recent technology (and are expected to continue to do so) – this makes the cloud model feasible
- A transformation of supply: “virtualisation” has become a commodity that businesses are willing to pay for. Companies have either adapted to meet their growing needs in computation and storage, or have outsourced this need to other companies

These three transformations lead to increased use, better networks and reduced costs. This then encourages increased use, and the cycle repeats

Cloud computing is becoming increasingly popular in science for many of the same reasons:

- A transformation in demand: many **new data sets** have the same basic requirements (scalability, data storage and processing)
  - New instruments are providing higher and higher resolution data
  - Numerical models are being run at higher and higher resolution



- This has led to dramatic increases in data size (Gb → Pb in less than a decade)
- We need new solutions to analyse these data

eNATL60 simulation by Ocean Next: <https://vimeo.com/307288779>

# Comparison with a 1° grid configuration...

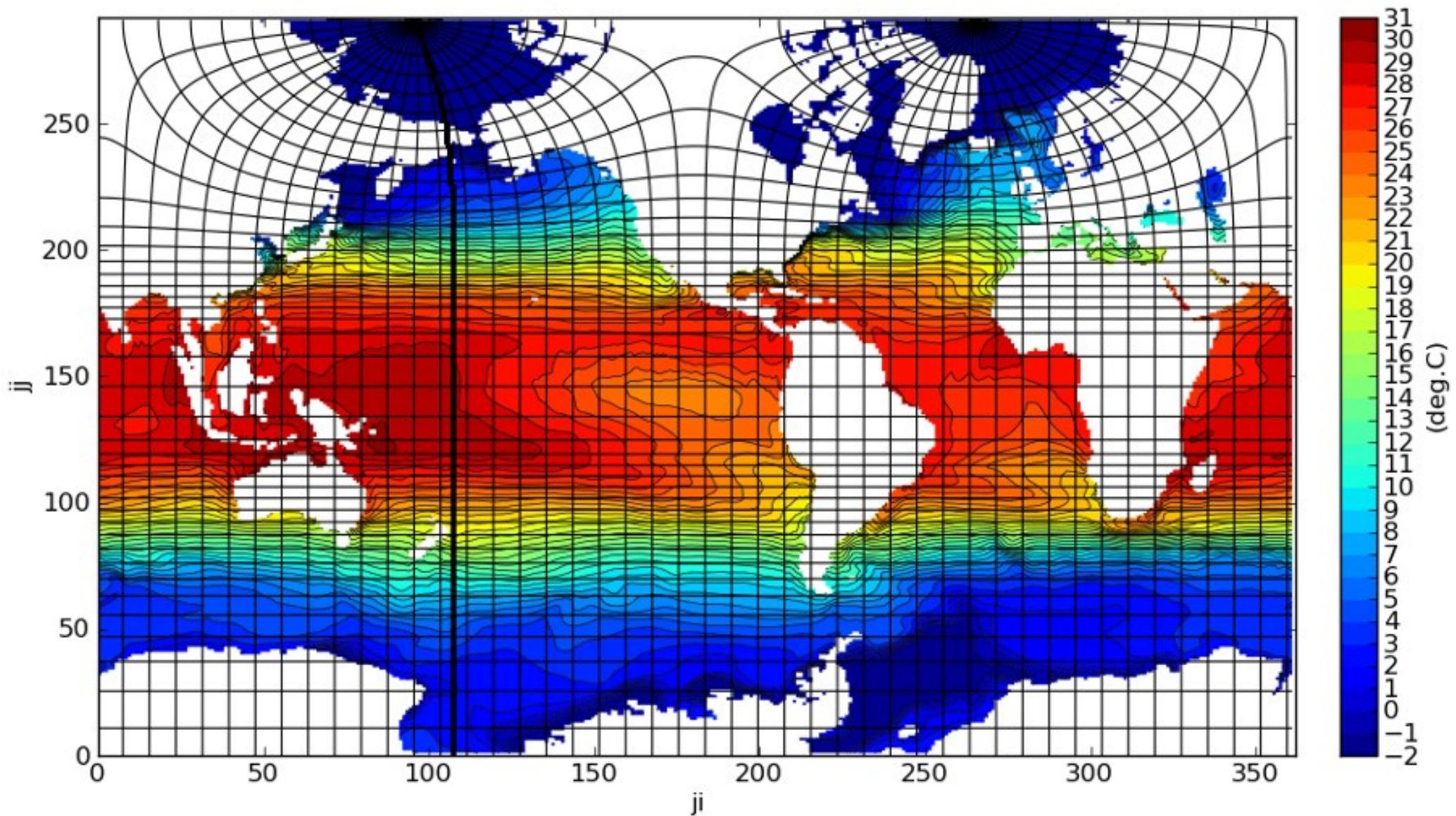
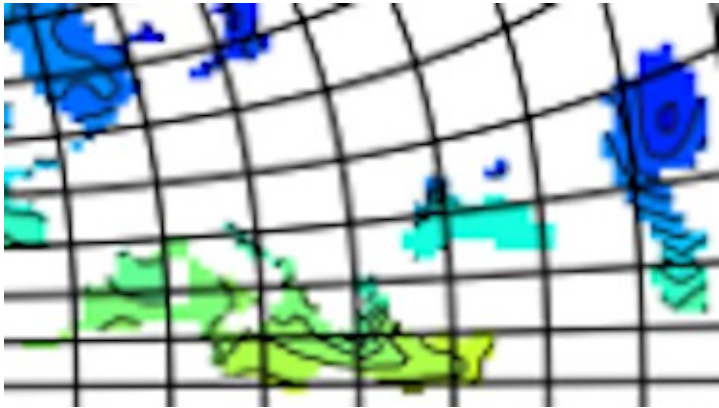


Figure by Laurent Brodeau:  
<https://brodeau.github.io/sosie/>

# Comparison with a 1° grid configuration...



In a 1° configuration, the Mediterranean Sea is described using 16 points in the vertical, and 42 points in the horizontal

In a 1/60° configuration, this would increase to 960 x 2520 !

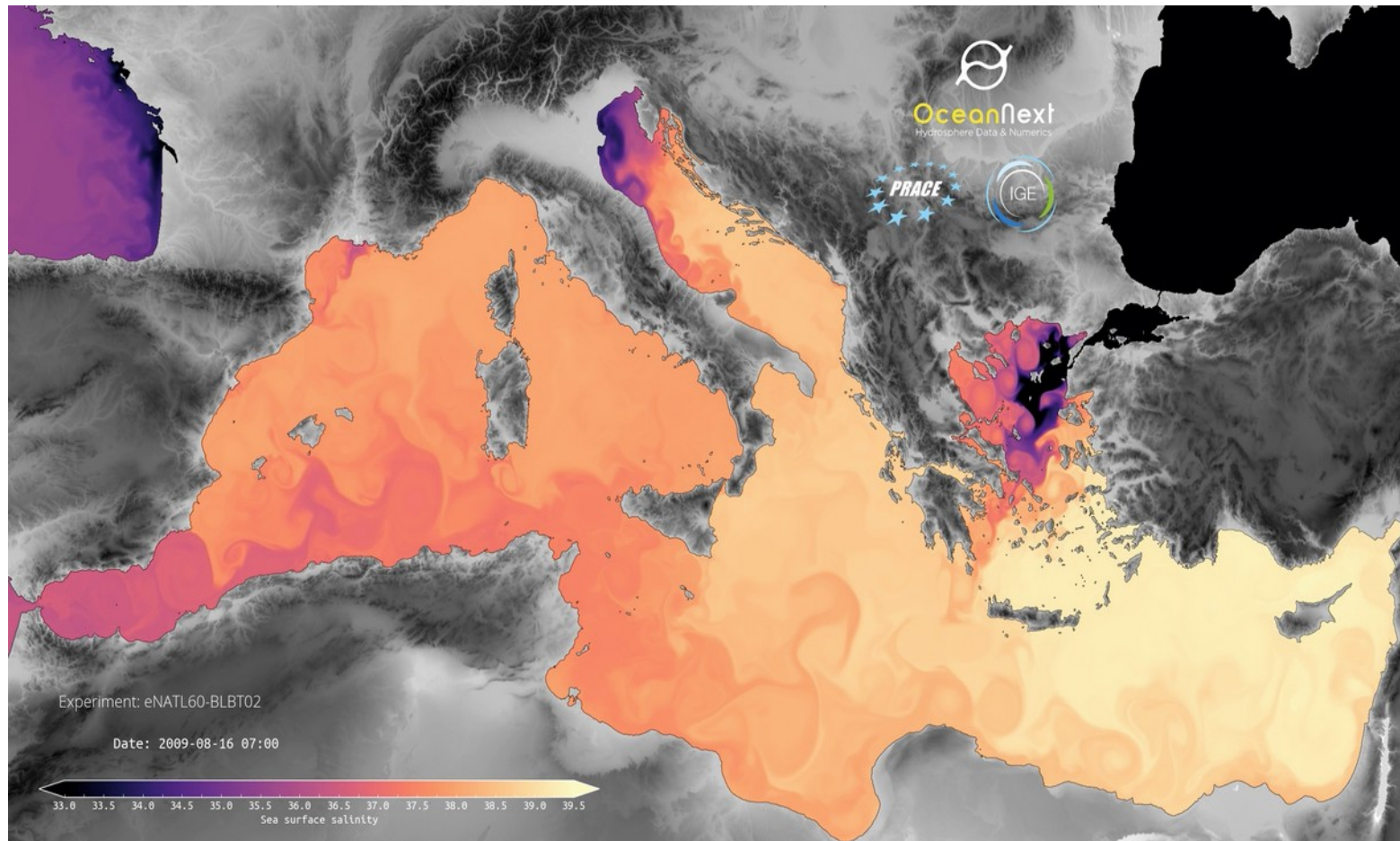
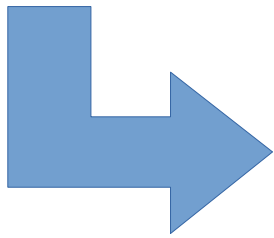
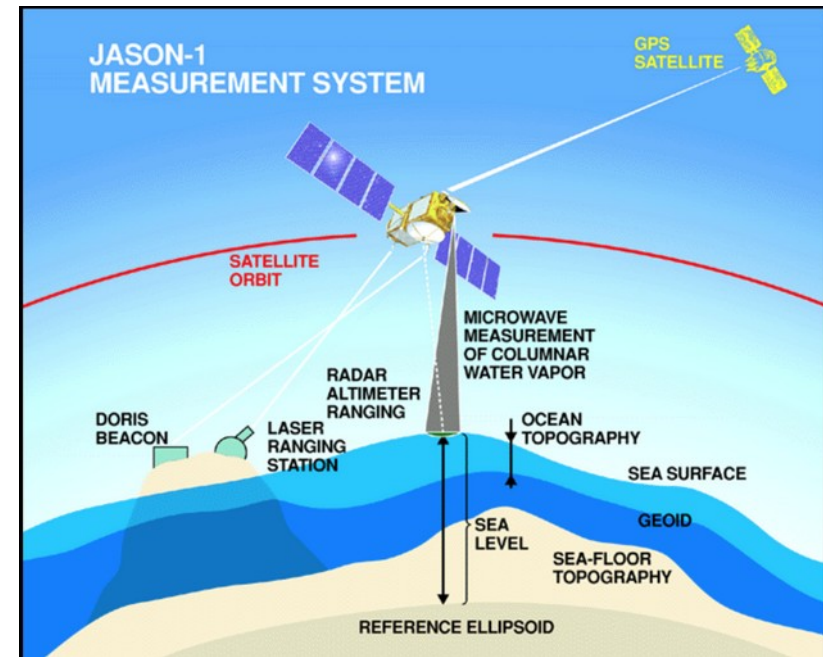


Image by  
OceanNext:  
[https://vimeo.com/  
318778679](https://vimeo.com/318778679)

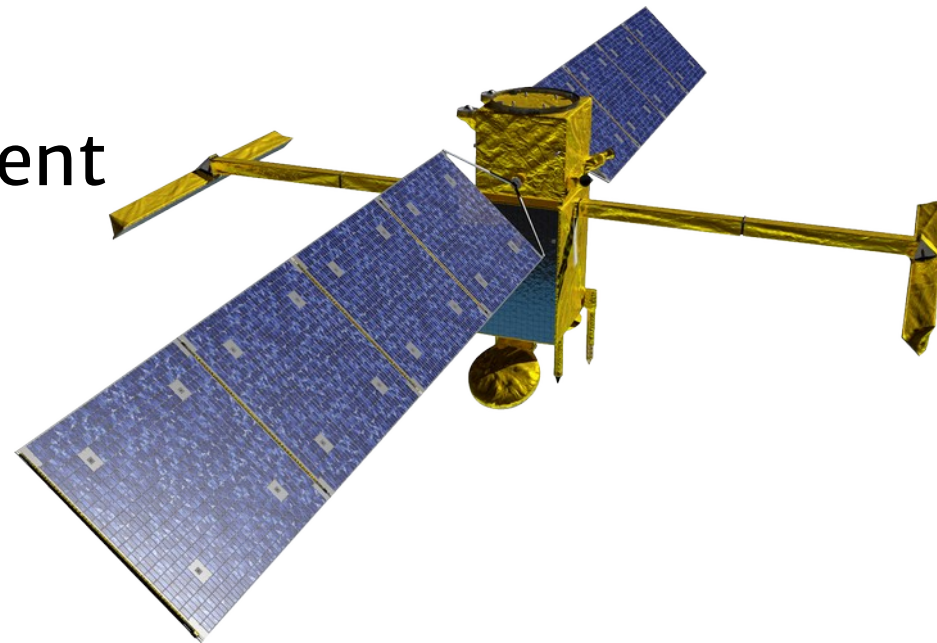
# The ever-increasing size of marine data

Another example:

- The first satellite altimeter (instrument used to measure sea surface height) was launched in 1992; it has a spatial resolution of  $0.25^\circ \rightarrow$  approx 28km at the equator
- The next-generation instrument that is expected to launch in 2022 will measure the sea surface height with a spatial resolution of 1km



By NASA/JPL, Public Domain



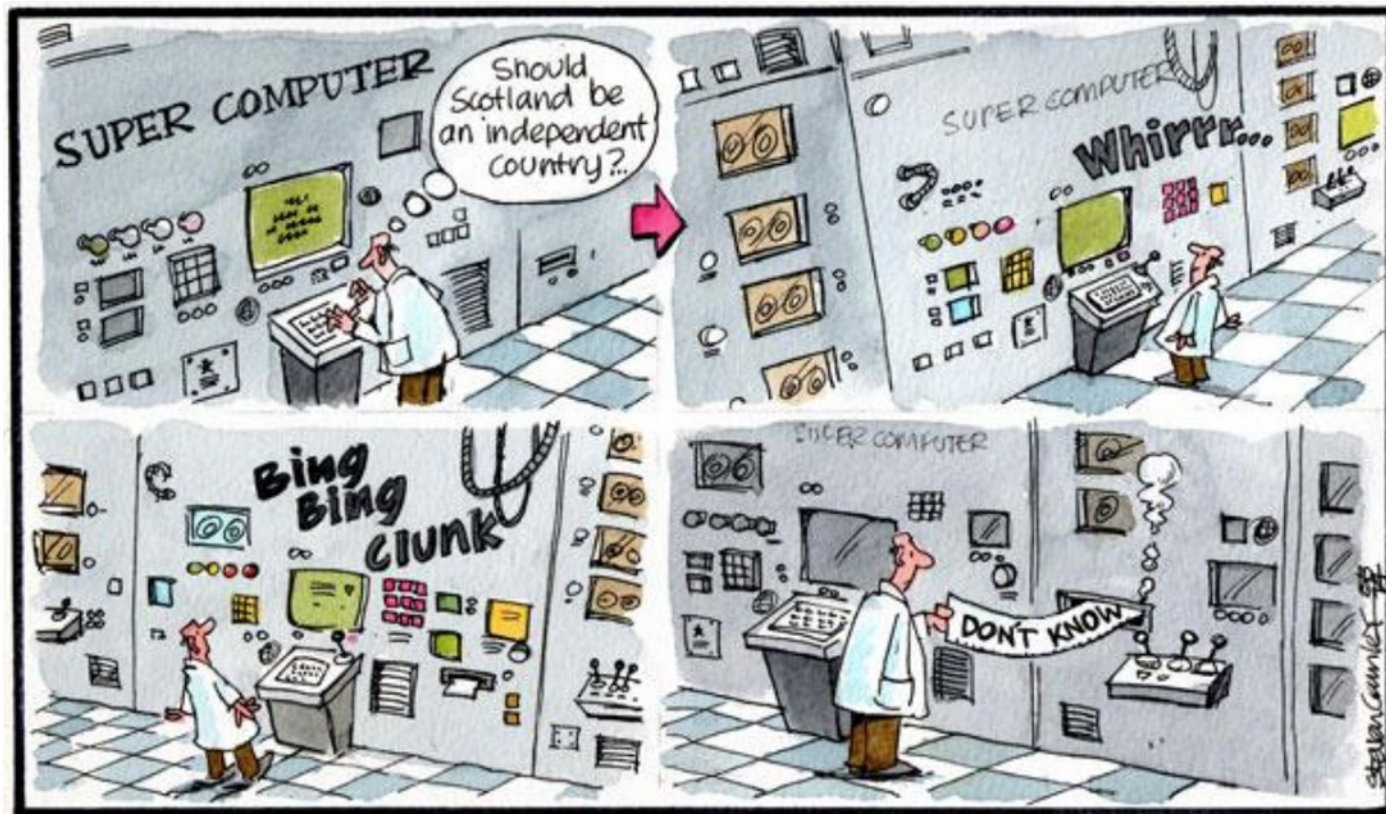


# The ever-increasing size of marine data

- New instruments and new model configurations are created in order to answer scientific questions that we can't answer with our existing data
- An important question in physical oceanography relates to the energy budget: where does the energy that drives the ocean circulation come from, where does it go, and how does it get transferred between different-sized flow features? To answer these questions, we need high resolution data, but...
- These new data sets are so large that we are still adjusting to find the best ways to work with them. They present difficulties in several respects: the quantity of raw data is much bigger than we are used to, the software that we historically used to analyse our data is not usually well adapted to analyse these data sets, and the machines that we usually analyse our data on are not powerful enough to store or to analyse such large volumes of data

# How can we work with big data?

- Numerical simulations are often produced on supercomputers. These are arrays of CPU processors that can communicate with one another, and a series of storage systems
- Historically, modellers would create their simulations on a supercomputer, and then transfer their data to a local server for analysis. Some institutions have their own supercomputers; otherwise, scientists might use national infrastructure for this.



# Mare Nostrum 4: Spanish supercomputer



Torre Girona chapel. Photo from Barcelona Supercomputing Center – [www.bsc.es](http://www.bsc.es)

# Mare Nostrum 4: Spanish supercomputer

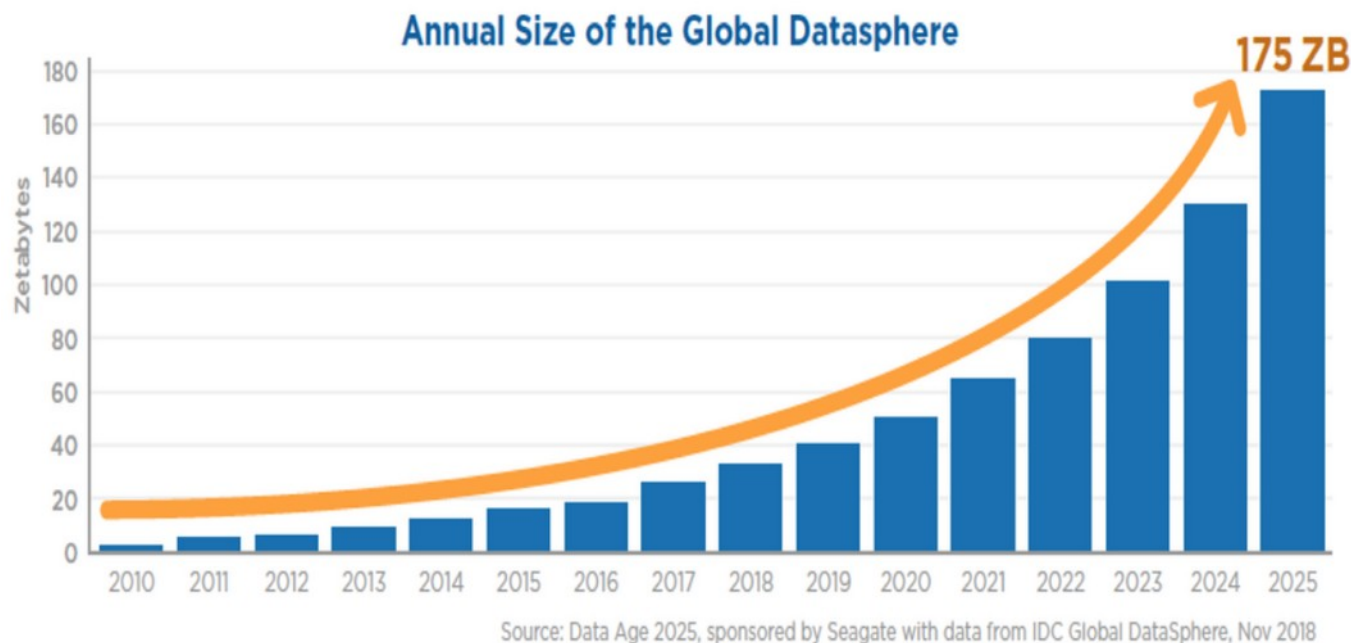


# How can we work with big data?

- With the increasing size of data, in some cases it is becoming unfeasible to perform analysis of data directly on a local machine. Either we need to work with smaller subsets of the data, or we need alternative ways to perform the analysis
- Possible options include:
  1. Use the supercomputer to perform the analysis of the data (for models, the data is left in place on the supercomputer's data storage system)
  2. Use cloud resources
- The increasing amount of data that is available has also driven interest in the application of data science techniques to marine science data
- The implementation of neural networks is normally done on a GPU: historically, scientists have not widely used these processors in their analyses, and so often the easiest way to have access is (again) through using shared resources, such as the cloud

# Will we need new ways to work with big marine data?

- It seems likely. But this is not a new phenomenon: for example, when  $1/4^\circ$  model configurations were invented, it seemed like a lot of data at the time! Scientists have always had to adapt, and both technology and our analysis methods always evolving
- However: it seems unlikely that we will continue with a model where we store (all) of our data on our local devices



NB: data size prefix order is:

Kilo ( $\times 10^3$ )

Mega ( $\times 10^6$ )

Giga ( $\times 10^9$ )

Tera ( $\times 10^{12}$ )

Peta ( $\times 10^{15}$ )

Exa ( $\times 10^{18}$ )

Zetta ( $\times 10^{21}$ )

So 1 Zb =  $10^{12}$  Gb!

# An example: the CMIP6 archive

Google Cloud

<https://cloud.google.com/blog/products/data-analytics/new-climate-model-data-now-google-public-datasets>

Blog Latest Stories What's New Product News Solutions & Technologies Topics



Keep up with the latest announcements from Google Cloud Next '21. Click [here](#).

DATA ANALYTICS

## New climate model data now in Google Public Datasets

Figure by  
Carbon Brief

**Shane Glass**  
Developer Advocate,  
Google Cloud

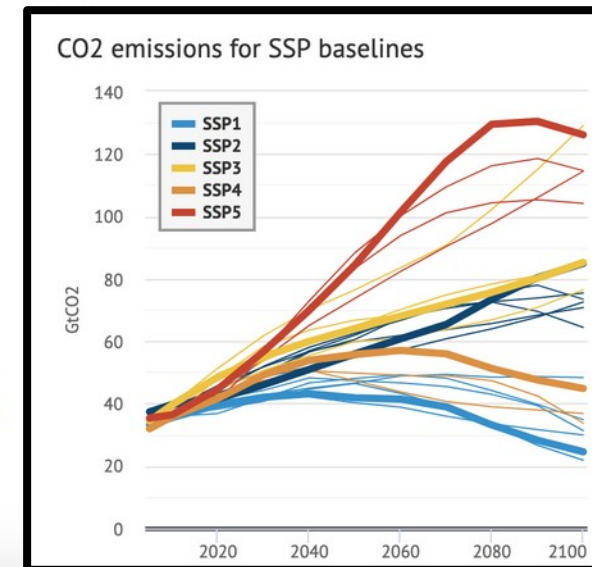
December 9, 2019

Try GCP

Start building on Google

Exploring [public datasets](#) is an important aspect of modern data analytics, and all this gathered data can help us understand our world. At Google Cloud, we maintain a collection of public datasets, and we're pleased to collaborate with the [Lamont-Doherty Earth Observatory](#) (LDEO) of Columbia University and the Pangeo Project [to host the latest climate simulation data in the cloud](#).

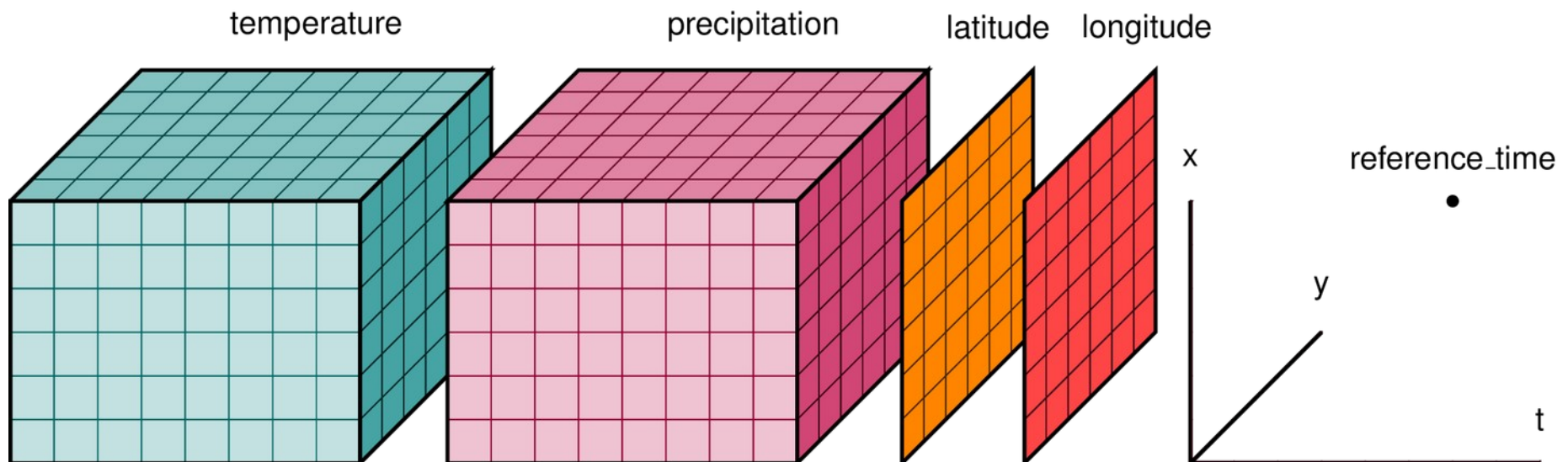
The [World Climate Research Programme](#) (WCRP) recently began releasing the Coupled Model Intercomparison Project Phase 6 (CMIP6) data archive



The Coupled Model Intercomparison Project phase 6 data are available using standard access methods (e.g. OPeNDAP) and also via cloud access. This provides a number of different possible ways to interact with this data.

# Data formats and cloud computing

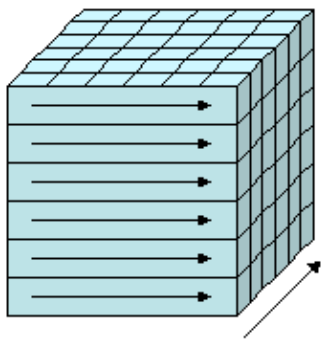
- A lot of marine data is stored in **NetCDF format**: this is the current “standard” format. A NetCDF file contains both **data and metadata**
- It is useful to have the metadata stored along with data because normally we need extra information to understand what all the dimensions on the data are, what they correspond to, what the units are, what variables are stored in the file...
- The person who has created the file can store all of this information along with the data itself – we don’t have to keep track of several files, with one for each variable and a corresponding description



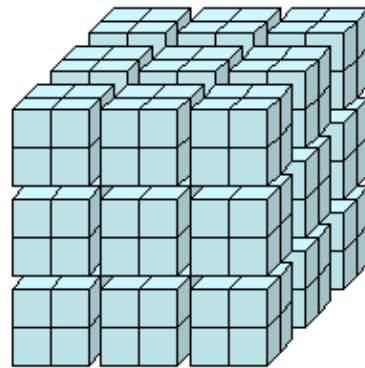


# Data formats and cloud computing

- Loading large amounts of data takes a long time. To help with this, in NetCDF files, data are stored in “chunks”. When the file is created, the data can be stored in a series of smaller rectangles (“chunks”) within the file. This can help us to read the data faster if the chunks are well-chosen, but...
- NetCDF was designed for file system use, not for cloud computing use. It has some performance issues when used in a cloud computing context because the storage system works differently



index  
order

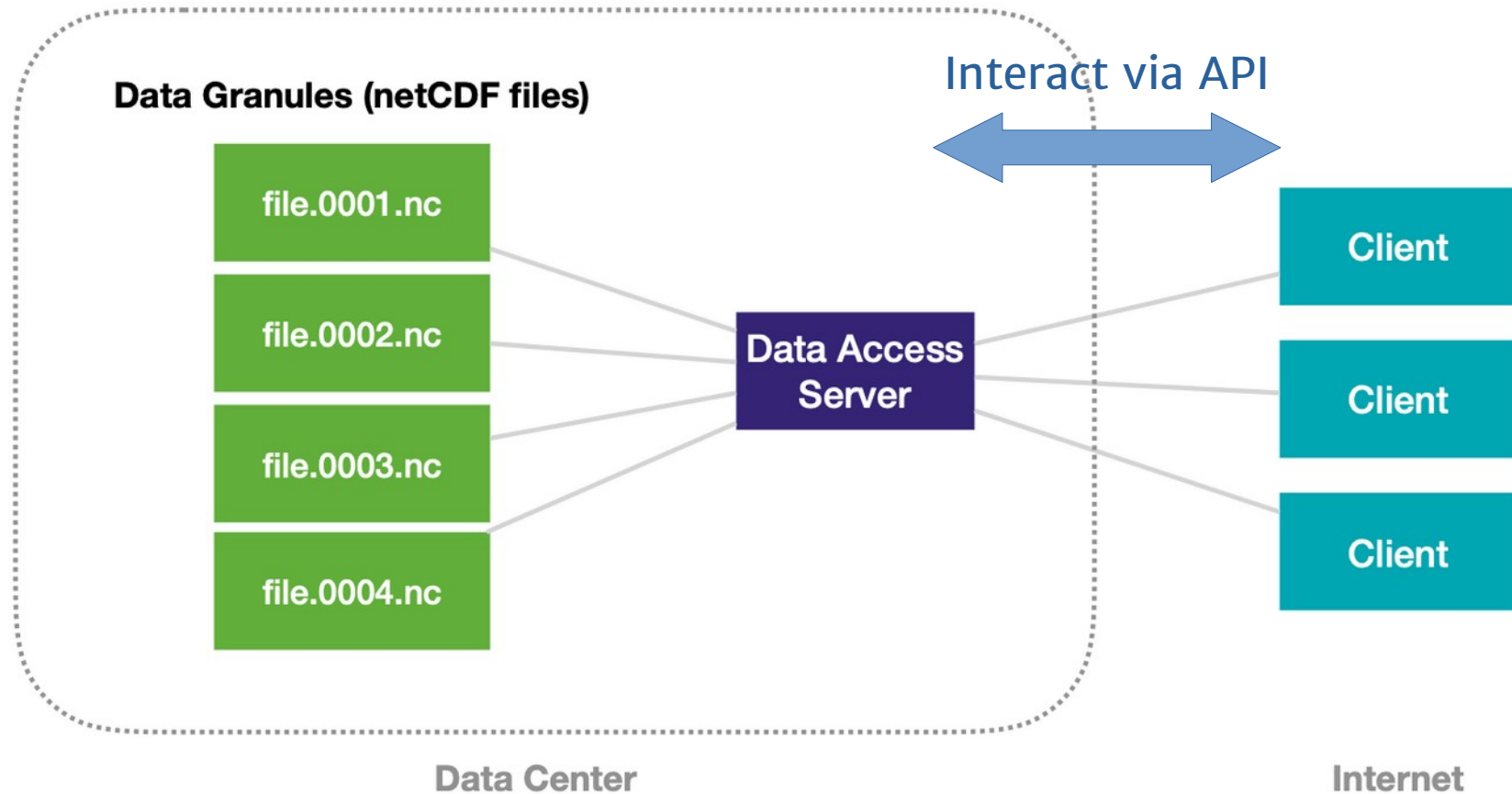


chunked

- The **Zarr format is a cloud-native format that uses the same data model as NetCDF, but adapted for efficient use with the cloud storage model**
- This format is starting to be adopted in the Earth sciences for use with very large data sets

# Accessing cloud-native format data

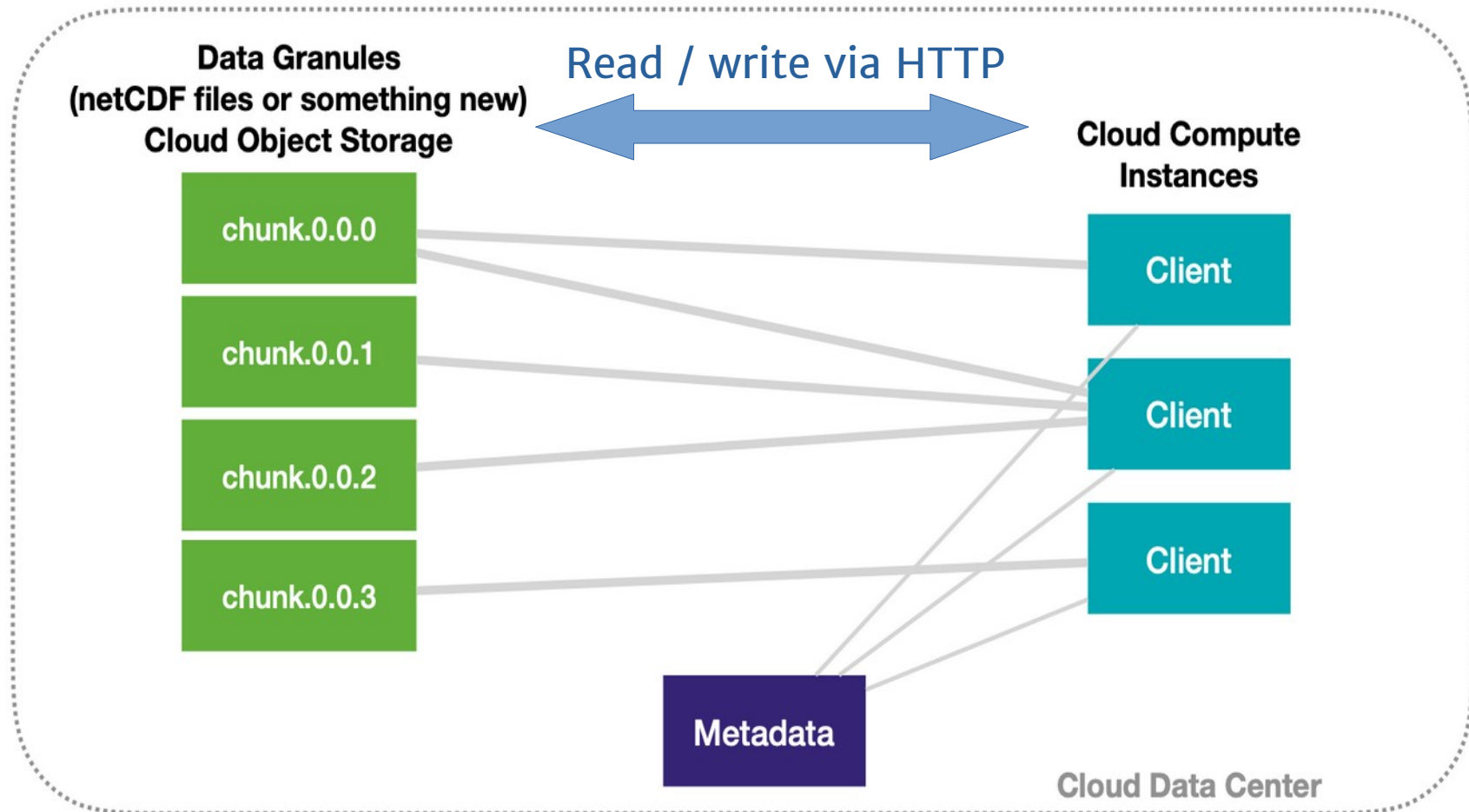
## Traditional Approach: A Data Access Portal



From Abernathey et al 2018: Beyond netCDF: Cloud Native Climate Data with Zarr and XArray

# Accessing cloud-native format data

## Direct Access to Cloud Object Storage



From Abernathey et al 2018: Beyond netCDF: Cloud Native Climate Data with Zarr and XArray

# Cloud computing summary

- Cloud computing has become increasingly popular over the past decade as the amount of data produced is growing dramatically and requires ever-increasing data processing power and storage
- The amount of marine data available is also growing dramatically. It seems likely that this will require new approaches to how we store and process data. Cloud computing may be a solution to this
- The architecture of cloud systems and file systems are very different. Formats that are well-adapted for use on file systems (e.g. NetCDF) are less well-adapted for use with cloud storage
- New formats and access methods are being developed for efficient access to structured data (= gridded, time series, etc) via the cloud. Note that NetCDF can still be used in the cloud – it is just less efficient.

# Useful tools for processing Earth science data in the cloud

- **Google Colab** (=colaboratory) is a product that allows anybody to write and run python code via a web browser, using Google's resources
- It is based on Jupyter: an open source project that allows you to interact with python via the web browser on your local machine or a local server
- It is nominally free to use, unless you have very intense needs (in which case it will crash; a paid service is also proposed in this case)
- It supplies access not only to CPU but also to GPU. This can make it a good way to get access to GPU if you want to start trying out machine learning techniques

# A Jupyter notebook...

3a. Standardising data and finding the area under...

localhost:8889/notebooks/3a. Standardising data and... Search

jupyter 3a. Standardising data and finding the area und... Last Checkpoint: 10/17/2021 (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Run Run Voilà Appmode nbdiff

## 3.1: Standardising data

When we standardise data, we transform it so that it has a mean of zero, and a standard deviation of 1. The transform is:

$$Z = \frac{x - \mu}{\sigma}$$

where  $x$  is our data,  $\mu$  is the mean, and  $\sigma$  is the standard deviation.

This transform can be applied to any data set: we are focused on the normal distribution here, but data that have other distributions can also be standardised using this same transform.

We will make some plots below to see the transform work:

```
In [ ]: # import libraries
import numpy as np
import matplotlib.pyplot as plt
plt.ion()
%matplotlib inline
```

```
In [ ]: # generate a sample of 5 random data points drawn from a normal distribution with
# mean = -3 and standard deviation = 5
x = np.random.normal(loc=-3, scale=5, size=5)
print(np.sort(x))
```

# And Google Colab...

TD1 organising and visualizing data.ipynb

File Edit View Insert Runtime Tools Help Last edited on September 19

RAM Disk Editing

## TD1: organising and visualizing data

### 1: Finding outliers in Argo data

You will analyse temperature and salinity data collected by [Argo](#) floats. These are instruments that drift independently in the ocean at a depth of 1000m. Every 10 days, they descend to 2000m, and then rise back up to the surface, collecting measurements of the temperature and salinity of the water column as they do so. Each set of these measurements (from 2000m depth to surface) is called a *profile*. They then return this information by satellite, before re-descending to 1000m and drifting for another 10 days.

0m

500

1 Float deployment

2 Descent to

3 Ascent: measuring Essential Ocean Variables

4 Data transmission

5 Starting next cycle

6 Argo data

7 Starting next cycle

# Binder



Reproducible, sharable, interactive computing environments

- Another service that allows you to run code in the cloud is Binder
- The Binder project allows other people to interact with code that you have written without having to run it on their own machine. You give it the link to your code, and it creates a link where other people can run your code
- It aims to improve reproducibility: there is a growing belief in science that a full description of a piece of work should include not only the final paper, but also the data and code needed to reproduce the analysis. Being able to re-run the analysis is part of this.



# Binder: an example

- A notebook from a data analysis class:

da1\_chapter\_1\_interactive\_figures X

https://hub.gke2.mybinder.org/user/closes-da1\_notebooks

jupyter da1\_chapter\_1\_interactive\_figures (unsaved changes)

Python 3 (ipykernel)

Trusted

Memory: 203.9 MB / 2 GB

## 1: Histogram generator

The following cell generates an interactive plot where you can change the mean value of the data, the standard deviation (stdev), the skew and number of points (npts) of the data set by moving the sliders. The right hand plot shows part of the random sample of data that is used to generate the distribution.

```
In [5]: from ipywidgets import interactive
from figure_functions import *
import matplotlib.pyplot as plt
%matplotlib inline
plt.ion()

iplot = interactive(plotpdf, mean=(-5.0, 5.01, 1.0), stdev=(0.1, 2.1, 0.2), skew=(-0.9, 0.92, 0.1), np
output = iplot.children[-1];
#output.layout.height = '350px'
iplot
```

mean

stdev

skew

# The Pangeo ecosystem (Python)

- Pangeo is a community of scientists, software and infrastructure developers working together to develop solutions to make it easier to use big data in the geosciences
- A Pangeo environment groups together a number of different Python tools that are useful in analysing big data, either in the cloud or on a supercomputer cluster
- From the [pangeo website](#):

## OUR GOALS

1. Foster collaboration around the open source scientific python ecosystem for ocean / atmosphere / land / climate science.
2. Support the development with domain-specific geoscience packages.
3. Improve scalability of these tools to handle petabyte-scale datasets on HPC and cloud platforms.

# The core tools:



Xarray



- Website: <http://xarray.pydata.org/en/latest>
- GitHub: <https://github.com/pydata/xarray>

xarray is a library that makes it easy to deal with multi-dimensional data: it can perform “standard” operations like finding the mean, the standard deviation, the maximum or minimum on large data sets in an efficient way. It also allows you to easily plot your data.

Dask



- Website: <http://dask.readthedocs.io/en/latest/>
- GitHub: <https://github.com/dask/dask>

Dask works with xarray to make it possible to work with very large data sets. The user defines the calculations that they want to perform, and then Dask manages the reading and calculation of the data in “chunks”, so that the calculation is possible given the resources available to the computer (= computer memory)

# The core tools:

Iris



## Iris

- Website: <https://scitools.org.uk/iris/docs/latest/>
- GitHub: <https://github.com/SciTools/iris>

Iris is similar to xarray: it is designed to work with common meteorology and oceanography data formats and also allows the data to be plotted easily

Jupyter



- Website: <http://jupyter.org/>
- GitHub: <https://github.com/jupyter>

Jupyter provides the interface between the user and the remote system

# Pangeo's vision for Earth science data analysis

## ON-DEMAND ANALYSIS-READY DATA

- **Too big to move:** assume data is to be used but not copied
- **Self-describing:** data and metadata packaged together
- **On-demand:** data can be read/used in its current form from anywhere
- **Analysis-ready:** no pre-processing required

From: <https://www.ecmwf.int/sites/default/files/elibrary/2018/18726-pangeo-ecosystem-data-proximate-analytics.pdf>

# Wrap up: how do we access data?

- Looking at the Pangeo vision of data analysis, and comparing it with where we started today, talking about web scraping, we can see that there is a full spectrum of possibilities in terms of how ready the data are for use, what preprocessing will be required, and what supplementary information will be available

WEB SCRAPING

~~ON DEMAND ANALYSIS READY DATA~~

Need to be moved:...

- ~~Too big to move~~: assume data is to be ~~used but not copied~~ ...to your local machine  
You do the...    ...and create the...    ...yourself    machine
- ~~Self-describing~~: ~~data and metadata packaged together~~  
...from your local machine
- ~~On demand~~: data can be read/~~used in its current form from anywhere~~

very unlikely    ...lots of...  
to be...

- ~~Analysis-ready~~: ~~no~~ pre-processing required

# Wrap up: how do we access data?

- Obtaining, cleaning up and transforming data are time-consuming and often tedious tasks. The examples that we have seen today go from “most effort required by the user” to “least effort required by the user”
- When dealing with small amounts of data, it is feasible to manage the processing yourself. Once you have very large amounts of data, automated systems become necessary
- There are a number of different models of data access depending on use cases, the size of the data, and what you want to do with it (both in terms of the subset that you want to work with, and the calculations that you want to perform)
- The increasing data volume associated with new observing technologies and high resolution numerical models will probably lead us to work in new ways in the future, perhaps involving heavier reliance on cloud-based systems. But scientists have always had to adapt to evolving technology: this is not an entirely new situation.